CRWR Online Report 04-04

Geographically Integrated Hydrologic Modeling Systems

by

Timothy Lee Whiteaker, PhD.

Graduate Research Assistant

and

David R. Maidment, PhD.

Principal Investigator

May 2004

CENTER FOR RESEARCH IN WATER RESOURCES

Bureau of Engineering Research • The University of Texas at Austin
J.J. Pickle Research Campus • Austin, TX 78712-4497

This document is available online via World Wide Web at
http://www.ce.utexas.edu/centers/crwr/reports/online.html

Acknowledgements

I would like to thank Dr. David Maidment of the Center for Research in Water Resources for his guidance and visions. I would also like to thank the GIS in Water Resources Consortium and the Texas Commission on Environmental Quality for their generous support of my research.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

Water resources engineers recognize the value in integrated modeling systems which take into account a variety of perspectives to produce solutions that sustain the integrity of the complete environmental system. In the past an engineer might have focused on a hydraulic simulation model to analyze the effects of channel straightening for flood conveyance purposes, whereas today's engineer might levy an increased importance to morphological and ecological impacts of channel modifications.

The need to integrate a variety of data sources and models leads to the concept of a hydrologic information system. A hydrologic information system, or HIS, organizes data in a manner which supports hydrologic modeling. An HIS also enables communication between simulation models and data, and even between different hydrologic models in the system. Recent advances in data management and information system technology have provided the tools necessary to construct an HIS that is both efficient and robust. The research presented in this dissertation defines techniques for creating geographically integrated hydrologic modeling systems with a focus on three types of integration: regional integration of subregional datasets, model integration at information exchange points, and the integration of geospatial features with processing engines. Case studies involving water quality, water supply, and floodplain mapping applications illustrate techniques for implementing the three types of integration. The research builds on the capabilities of ArcGIS 9 geographic information systems software and the ArcGIS Hydro data model.

1.1 Background

Hydrology is the study of the state and movement of water on or below the earth's surface and in the atmosphere. A hydrologic model approximates a real hydrologic system through a physical representation of the system, or in an abstract fashion using equations (Chow, et al., 1988). The research outlined in this dissertation utilizes this type of model, and from this point on, the terms 'hydrologic model' or 'simulation model' refer to an abstract hydrologic model that is implemented using a computer. Note that a simulation model is different from a data model. Whereas a hydrologic simulation model represents processes, a hydrologic data model represents properties and relationships among classes of geospatial and temporal hydrologic data. In other words, a hydrologic data model describes the water environment, while a hydrologic simulation model describes how water moves through the environment.

Hydrologic models transform input data about the hydrologic system using a set of equations or numerical methods to produce output data that provide useful information about the state of the system or how the system changes over time. Inputs for models are either created through a model's graphical user interface (GUI), or are fed to the model through files (such as ASCII text files) or in-memory functions. Outputs can either be read directly (e.g., in the form of tables) or displayed graphically using a model GUI or other software.

Often the spatial and temporal data components of a hydrologic model are best handled using a GIS. A Geographic Information System, or GIS, enables the creation, analysis, storing, and display of geographically referenced information. While first used primarily for display purposes, advances in computing power and GIS technology have led to the evolution of powerful mapping and analysis capabilities within the GIS (Goodchild and Kemp, 1990). A GIS may be used to process raw input data to create necessary input parameters for simulation models. For instance, STATSGO soil data may be overlain with land use zones to produce curve numbers for a set of watersheds. A GIS is also useful for providing quick and valuable interpretation of model outputs through graphical display. For instance, output from a hydraulic model can be used to create a 3-dimensional representation of inundation zones and the buildings that would be affected by a given flood wave.

The ArcGIS Hydro data model (Arc Hydro) provides a framework for storing geospatial and temporal information in a GIS in a manner that supports hydrologic simulation modeling. The goal of Arc Hydro is to bring the GIS world and the water resources world closer together through a tighter integration between GIS data and hydrologic models. This research utilizes Arc Hydro to facilitate the geospatial integration of data and simulation models.

This research makes use of the latest GIS software from ESRI, ArcGIS 9. ArcGIS 9 introduces an environment for assembling workflow models of geoprocessing tasks. Standard ArcGIS tools or custom tools may be chained together in a workflow model to perform complex work. Thus, three types of models are utilized in this research: simulation models, data models, and workflow models. A description of each of these models is summarized in Table 1.1

Model Type	Description	Example
Hydrologic Simulation Model	Computational model simulating hydrologic processes on a computer	HEC-HMS
Hydrologic Data Model	Provides a data structure to represent properties and relationships among classes of geospatial and temporal hydrologic data	ArcGIS Hydro Data Model
Workflow Model	ArcGIS 9 Geoprocessing model organizing individual tasks into workflow sequences to perform complex tasks	NEXRAD to Flood Polygon (presented in Chapter 5)

Table 1.1 Summary of Model Types Used in This Research

1.2 HYDROLOGIC INFORMATION SYSTEMS

The use of a variety of data types, models, and software to create an integrated hydrologic modeling system requires an information system capable of managing the communication of information between the different components in the system. The Consortium of Universities for the Advancement of Hydrologic Science Inc. (CUAHSI), a collaborative effort between 78 American universities and research institutions to advance hydrologic science, defines such a system as a **Hydrologic Information System** (Graham et al, 2002). As hydrologic data become more voluminous and widely available, and modeling systems incorporate a greater variety of individual models to simulate integrated hydrologic responses, the technology of information systems takes on a greater relevance to hydrologic modeling. The development of a hydrologic information system is not only guided by programming practices and database technology, but also by the science of hydrology, which requires standardized data structures, metadata descriptions, and both temporal and spatial interpolation schemes in order for the data to be useful across the range of hydrologic models and spatial scale.

Hydrologic Information Systems provide a means of integrating large storehouses of geospatial and temporal data with simulation models. While prototype HIS have been implemented for years, the effort to formally define the science of HIS is relatively new (Graham et al, 2002). Thus, a common standard for developing an HIS has yet to be adopted.

This research contributes to the development of that standard by defining an HIS structure, which uses a GIS to store geospatial and temporal data for hydrologic modeling. The geodatabase utilizes the Arc Hydro schema to adapt GIS data, which is typically cartographic in nature, to a more model-oriented design. Justification for using Arc Hydro includes:

- Arc Hydro was designed to store hydrologic GIS data in a manner conducive to data export for model use. Therefore, much research and investigation was performed to determine which data features (e.g., watersheds), constructs (e.g., geometric networks), and attributes (e.g., HydroID) are important to hydrologic modeling and should be included in the geodatabase.

- Arc Hydro has been adopted as a data standard for storing hydrologic information in state and national datasets, including the National Hydrography Dataset, USGS Streamstats, and the Water Availability Model for Texas. The South Florida Water Management District and the Hydrologic Engineering Center are also considering adopting extensions of Arc Hydro as data standards.
- Several toolsets have been developed to operate on Arc Hydro data, including the Arc Hydro tools and the WRAP Hydro tools, a toolset designed specifically to preprocess Arc Hydro data for use in the WRAP simulation model.
- Several efforts have already been undertaken to import and use data in the Arc Hydro structure, providing insight and experience to the question of "How do we use Arc Hydro?", including Gopalan (2003), Zoun (2003), and Stone (2001).

Thus, this research contributes to the advancement of hydrologic information systems by exploring means of integrating models and data through standard data structures and widely available commercial software. In fact, the development of the hydrologic modeling systems described in this research would not be possible without a hydrologic information system to facilitate information exchange between components in the system.

1.3 Interface Data Models

A key concept developed through this research is the **Interface Data Model**. An Interface Data Model provides a two-way link between a GIS and a hydrologic simulation model. Hydrologic models typically require data describing the properties and connectivity of features that transmit or affect the movement of water throughout the landscape. Thus, the nature of a hydrologic model is not only governed by the fundamental hydrologic principles, equations, and numerical methods upon which the model is based, but also by the quality, precision, and availability of the data to be used in the model. Increases in computing power, data collection, and data distribution through means such as the Internet allow hydrologists to use more distributed models with increasingly large spatial data requirements.

While Arc Hydro provides a good starting point for adapting data to hydrologic models, Arc Hydro cannot accommodate every aspect of geospatial information that every model requires. Therefore, an **Interface Data Model** is created, which defines a GIS data structure that includes information required by a particular simulation model. A separate Interface Data Model is created for each specific simulation model. Therefore, if a geodatabase supported inputs for three independent simulation models, then three Interface Data Models would be developed within the geodatabase. In some cases, such as with HEC's HMS and RAS models, components of Interface Data Models may overlap (i.e., the DSS time series component in the HEC models.) An Interface Data Model defines a structure for data storage, much like Arc Hydro. An Interface Data Model does not contain the simulation model itself, but rather the information that the simulation model requires prior to execution, as well as outputs from the simulation model after execution. Thus, an Interface Data Model facilitates two-way communication between GIS and a simulation model.

To further illustrate the concept of Interface Data Models, first consider the previous way that geospatial integration of GIS with models was accomplished. As evidenced in applications such as Hudgens (1999), distinct geodatabases and preprocessors were developed for each model application. The role of the GIS was predominantly as a preprocessor for geospatial inputs into the model. Thus, a **Preprocessing Data Model** was developed for each application. A preprocessing data model contains the structure necessary to store the geospatial inputs required by the hydrologic simulation model. The direction of communication between the GIS and the hydrologic simulation model is typically one-way, from the GIS to the model.

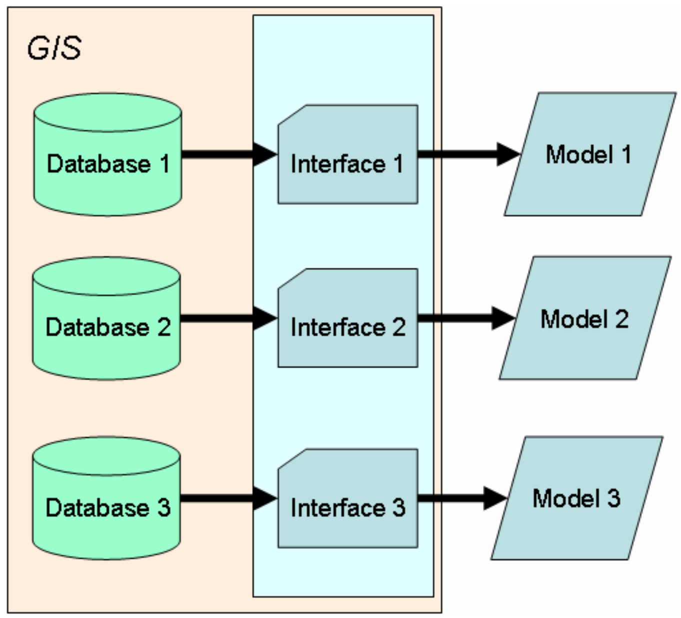


Figure 1.1 Preprocessing Data Model

While often serving the purpose of linking to a specific model quite well, applications built with preprocessing data models are difficult to extend such that other simulation models can take advantage of geospatial data development for the initial model. In other words, the geodatabases are not compatible between different model applications.

With **Interface Data Models**, the components needed to describe a given model's inputs, properties, and outputs are stored in a GIS, thus facilitating the two-way exchange of information between the GIS and simulation model. Because the Interface Data Model is designed to work specifically with the simulation model, communication between the Interface Data Model and the simulation model is efficient and manageable.

Furthermore, by storing model data in the geodatabase, and by recognizing common hydrologic features in different simulation models as extensions of core Arc Hydro features, data may be shared among multiple simulation models. By using Arc Hydro as a data standard for core hydrologic features, a single source of hydrologic geospatial data, such as watershed and stream network information, can be used in multiple simulation models. Additionally, the output from one simulation model (Model A) may be used as input into another model (Model B), by bringing the data into an Interface Data Model (for Model A), through Arc Hydro, and then through another interface model (for Model B.)

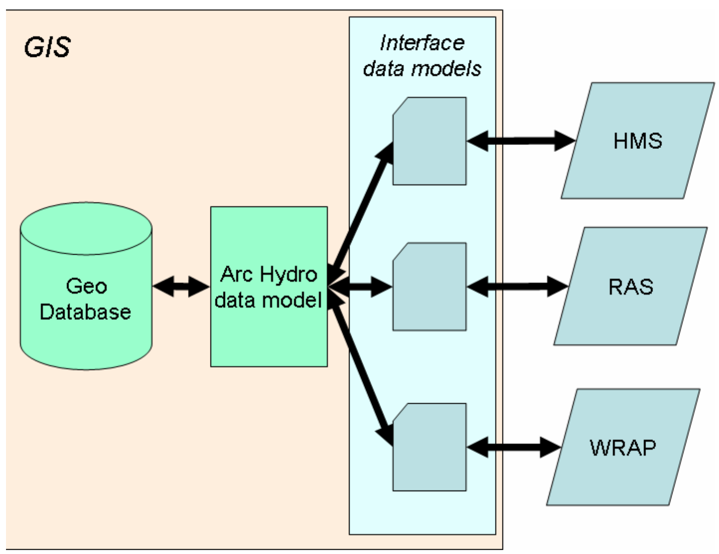


Figure 1.2 Multiple Interface Data Models for HMS, RAS, and WRAP

In summary, Interface Data Models provide a means of storing geospatial data for model input, storing model output, and sharing data between Interface Data Models through Arc Hydro, while still maintaining the autonomy of simulation models.

1.4 Problem Statement and Motivation

The concept of geospatial integration can be applied to both the integration of spatial data, and the connection of spatial and temporal data with simulation models. Several factors have dramatically increased the potential and need for geospatial integration in the past few years, including:

- Increased quantity and quality of hydrologic data through technologies such as LIDAR (Kidner and Smith, 2003) and funding for data collection
- Increased availability of data through the Internet
- Recognition of the importance of engineering solutions that integrate a variety of model types (e.g., hydrologic, hydraulic, ecological, etc.)
- Advances in computer technology
- Advances in programming techniques
- Increased power and sophistication of GIS software

However, certain obstacles stand as a hindrance to geospatial integration. For example, the high resolution of LIDAR data prevents straightforward raster analyses of that data over large areas, simply because today's computers cannot process such large datasets. Similarly, agencies that manage large river basins are faced with the challenge of how to incorporate finer resolution digital elevation models (DEMs) into watershed analysis routines. While an agency may desire to use 10-meter DEMs to delineate watersheds, this would require processing raster datasets nine times as large as datasets based on 30-meter DEMs. In some cases, the scope of analysis may be so large that even 30-meter DEMs prove cumbersome. The Texas Commission on Environmental Quality (TCEQ) encountered this problem when attempting to create Water Availability Models (WAM) covering river basins spanning the state of Texas (Figurski, 2001). Establishing a methodology for processing WAM GIS datasets is one of the motivations for the author's research.

Even when data management and geospatial analysis solutions have been attained, the challenge of integrating the data with models still remains. Hydrologic simulation models have existed for decades. The routines to perform hydrologic computations are well understood, and the models themselves have been validated through many years of use. However, most models are designed to operate solely as autonomous units with a specific focus, such as river channel hydraulics. Little consideration is given as to how the model, or data from the model, may interact with other models or databases. Each model typically has its own format for input and output data, which may be difficult to share with other models that could also make use of the same data. So while information from a hydrologic model might be useful in a hydraulic model, no method may exist to port the data between the two models other than tedious manual data entry. The San Antonio River Authority is working with the University of Texas at Austin to develop a seamless application that processes rainfall data to produce flood inundation polygons, by performing GIS operations and calling HEC-HMS and HEC-RAS hydrologic simulation models. This effort is another motivation for the author's research.

In some cases, it is more desirable to attach simple behavior to hydrologic features in a GIS, rather than call a sophisticated hydrologic simulation model.

For example, Zoun (2003) applied Arc Hydro to GIS data for Galveston Bay, Texas, in order to simulate water quality loadings to the bay. While Arc Hydro was useful in organizing and preprocessing the hydrologic data for the watersheds draining to the bay, no automated mechanism was available to simulate bacterial loadings and transport (which included a 1st order decay rate) along streams in the watersheds. The connectivity between hydrologic features was established through the Arc Hydro schematic network, yet the schematic features possessed no behaviors to govern the conveyance of loads through them and the interactions with adjacent features. A third motivation for the author's research is to develop a means of associating hydrologic behavior with features in an Arc Hydro schematic network.

As geographic information systems increasingly become the mechanism for data management, analysis, and mapping of water resources data, it is clear that a great benefit lies in using a GIS to facilitate the integration of geospatial data and simulation models. This research focuses on means of achieving that integration using Arc Hydro and the latest GIS software.

1.5 Objectives

This research addresses the TCEQ's difficulties in processing large raster datasets for the WAM effort by observing that a large regional dataset may be divided into hydrologically distinct subregions, whose size is conducive to raster analyses. GIS tools already exist for subdividing a large watershed into subwatersheds. The challenge lies in integrating the results of subregional analysis to produce valid results for regional analysis.

This research also explores methodologies for integrating geospatial and temporal data with mechanisms for simulating hydrologic behavior. One type of integration involves the connection of external hydrologic simulation models through Arc Hydro and a GIS. Another type of integration links individual hydrologic features in a GIS directly with a processing engine which defines how that feature will simulate hydrologic behavior.

Thus, this research provides answers to the following three questions:

1. How can subregional datasets be integrated into large regional datasets?
2. How can GIS and Arc Hydro be used to integrate hydrologic and hydraulic simulation models?
3. How can individual features in a GIS be directly integrated with processing engines to simulate hydrologic behavior?

The answers to these questions are framed within the context of ESRI's ArcGIS 9 software, Arc Hydro, and a selected set of simulation models and scenarios.

1.6 Contributions From Research

This research provides several toolsets, applications, and advancements to knowledge in water resources engineering. These contributions include:

Arc Hydro Tools - The Arc Hydro Toolset contains tools for creating Arc Hydro data and populating the attributes of Arc Hydro classes. The author developed the prototype Arc Hydro Toolset, which was later modified and extended by ESRI to form the official Arc Hydro Tools. The Arc Hydro tools populate the attributes of and perform analyses with the Arc Hydro data model, which facilitates the integration of geospatial and temporal data in a GIS with hydrologic simulation models. The latest toolset may be downloaded from http://www.crwr.utexas.edu/giswr/hydro/ArcHOSS/Downloads/index.cfm.

Raster-Network Regionalization Technique - The Raster-Network Regionalization Technique provides a methodology for integrating subregional datasets into large regional datasets, so that raster processing may be performed at the subregional level, and results may be integrated at the regional level using vector river networks. The Raster-Network Regionalization uses **watersheds as processing units** for attribute accumulation, which provides a much larger cell size (and thus faster computation time) than when working from raster datasets, without compromising the spatial variability of data required to characterize watershed characteristics for important points on the stream network.

The author developed the tools for performing the Raster-Network Regionalization Technique, including network-based accumulation and consolidation tools for calculating watershed parameters for key junctions in the stream network. The methodology for the technique initially evolved from a combined effort from Melissa Figurski and the author, with subsequent development by Hema Gopalan and the author. More information about the

Raster-Network Regionalization Technique can be found at http://www.crwr.utexas.edu/reports/2003/rpt03-3.shtml.

WRAP Hydro Data Model - The WRAP Hydro data model is a preprocessing data model, which defines a structure for storing geospatial data so that hydrologic parameters may be calculated for use in the WRAP hydrologic simulation model. The WRAP Hydro data model was primarily developed by Hema Gopalan, with assistance from the author. The WRAP Hydro data model may be downloaded from http://www.crwr.utexas.edu/gis/gishydro03/WRAPhydro/WRAPhydro.htm.

WRAP Hydro Tools - The WRAP Hydro Toolset contains tools for processing geospatial inputs in order to calculate hydrologic parameters required by the WRAP hydrologic simulation model. The WRAP Hydro Tools were developed entirely by the author, and include full Windows style help documentation. A key idea incorporated into the WRAP Hydro tools is that features of **point**, **line**, or **polygon** geometry can be used as source features for **watershed delineation**. Using line or polygon features as opposed to point features (which is the traditional way of delineating watersheds) provides a larger outlet zone for watershed delineation, helping to ensure that the intended watershed boundaries are actually delineated. The WRAP Hydro Tools may be downloaded from http://www.crwr.utexas.edu/gis/gishydro03/WRAPhydro/WRAPhydro.htm.

Interface Data Model - An Interface Data Model provides a GIS data structure which supports a particular hydrologic simulation model, and allows for two-way communication between a GIS and the simulation model. The concept of an Interface Data Model was developed by Dr. David Maidment and several team members, including the author, at the Center for Research in Water Resources at the University of Texas at Austin. Oscar Robayo, Dan Obenour, Jon Goodall, and Gil Strassberg have developed Interface Data Models for HEC-RAS, HEC-HMS, SPARROW, and MODFLOW, respectively. Examples of Interface Data Models may be found at http://www.crwr.utexas.edu/gis/gishydro03/GISHydro2003.htm.

Exchange of Data at Information Exchange Points - This concept allows simulation models to be integrated through a GIS by identifying discrete locations in space where models may exchange time series data. This concept is used by the NEXRAD to Flood Polygon Model to integrate the HEC-HMS hydrologic simulation model with the HEC-RAS hydraulic simulation model.

NEXRAD to Flood Polygon Model - This ArcGIS 9 workflow model processes NEXRAD rainfall data with terrain data to produce flood inundation polygons. The model incorporates GIS processing, HEC-HMS, and HEC-RAS into a seamless application. The model was developed by Oscar Robayo and the author. This model may be downloaded from the San Antonio River Authority at http://www.sara-gis-tx.org/metadataexplorer/explorer.jsp.

Schematic Processor - The schematic processor associates processing engines with Arc Hydro schematic features in a GIS to simulate hydrologic behavior. Each processing engine defines a particular type of hydrologic behavior. Different processing engines are associated with different features, depending on the type of the feature. The schematic processor was developed entirely by the author. Several processing engines used by the schematic processor have also been developed to support water quality analysis. These processing engines were developed by Jon Goodall and the author. The schematic processor may be downloaded from http://www.crwr.utexas.edu/gis/gishydro03/Schematics/SchematicNetwork.htm.

1.7 DISSERTATION OUTLINE

This dissertation is divided into seven chapters. The first chapter provides some background information about hydrologic modeling and Geographic Information Systems, followed by an overview of the objectives for this research. The second chapter reviews literature relevant to the proposed research. The third chapter reviews existing programming practices and technologies that provide useful insight in developing a hydrologic information system. The fourth chapter presents a method of integrating subregional geospatial data for watershed analysis using the stream network as the backbone for parameter calculations. The fifth chapter describes a technique of exchanging time series between models at information exchange points. The sixth chapter presents a method of directly integrating individual features in a GIS with processing engines, to simulate behavior in a schematic network of connected features. The seventh chapter discusses conclusions and recommendations based on this research.

Chapter 2 Literature Review

This chapter provides a review of research and applications related to the integration of geospatial data and simulation models.

2.1 Survey of Literature Regarding Geospatial Integration

2.1.1 GIS and Model Integration

A GIS can be a valuable tool in support of hydrologic and hydraulic modeling. Several applications have been developed over the years covering a variety of water resources concerns, including water quality (Prisloe Jr. et al, 2000; Yoon, 1996) and floodplain mapping (Robbins and Phipps, 1996; Correia et al, 1999; Koussis et al, 2003).

A GIS manages large volumes of geospatial data, such that distributed parameters can be used in a simulation model rather than lumped parameters (Vieux, 2001; Gao, Sorooshian, and Goodrich, 1993; Ogden et al, 2001). This can result in a more accurate depiction of the reality that the engineer is attempting to model. However, as de Roo (1998) points out, if the GIS data contain errors or do not translate correctly into model inputs, a distributed model may perform no better than a lumped one. Loague and Corwin (1998) reaffirm this, stating that as one integrates GIS and models, one not only has to worry about model error (the inability to simulate something correctly), but also data accuracy and parameter calculation (e.g. having enough data points to produce an accurate surface).

As a GIS and simulation models are designed separately, the type of data in a GIS may not conform to what the simulation model is expecting. Scale, precision, data structure, and data meaning are among the domains in which error could arise during communication between the GIS and the simulation model (Vieux, 2001). Roberts and Moore (1998) observed the importance a data model for easy data query and retrieval in decision support systems. A data model defines a standard data structure, and provides more concrete data meaning for a GIS application. Maidment (2002) developed a data model for water resources features, called Arc Hydro, to assist in GIS and model integration.

Charnock, Hedges, and Elgy (1996) describe two levels of GIS and simulation model integration. The first combines GIS and models through tight integration, with the two components communicating directly with each other. This results in large development costs, but typically requires less expertise from an end user. The second links GIS and models through communicating programs or bridges, in which the model and GIS programs are executed separately and simply share data through the bridge. Storck et al (1998) provides an example of this type of integration by using a GIS as a pre- and post-processor for the Distributed Soil-Hydrology-Vegetation Model (DHSVM), with information being exchanged between the GIS and DHSVM through binary or ASCII files. This approach requires less development time, but requires more expertise from the end user, as the user must be aware of the problems in matching the model's data requirements and representation of reality with what's available in the GIS (Charnock, Hedges, and Elgy, 1996). Again, the Arc Hydro data model

(Maidment, 2002) ameliorates this problem by providing a GIS data structure adapted for hydrologic and hydraulic modeling. Kopp (1996) further illustrates the importance in using data models, as a robust data structure alleviates some of the difficulties in developing data bridges. Kopp also observes that the connection of GIS and hydrologic simulation models through bridges is the most common type of integrated application that has been developed.

Clark (1998) observes that the emerging view of the role of a GIS is that of a database support and analysis system. The GIS is one of several applications, such as hydrologic simulation models, which support a central database of information. This setup is best supported by the use of bridges, as several applications must interact through the central database or with each other.

The research presented in this dissertation utilizes bridges to transfer geospatial and temporal data between a GIS and hydrologic simulation models. Chapter 5 describes the use of bridges at discrete information exchange points to transfer information between HEC-HMS and HEC-RAS through the GIS. Each bridge is custom-built for each simulation model. Chapter 6 describes a schematic network processor that operates at the feature level, which exchanges attribute information from schematic features by conforming to set a transfer rules prescribed by the schematic processor. This approach is different than that used in Chapter 5, in that the bridge used by the schematic processor remains the same, and that each processing engine must be written to exchange information using that bridge.

2.1.2 Toolboxes of Hydrologic Processors

Charnock, Hedges, and Elgy (1996) recommend focusing simulations models on the specific hydrologic and hydraulic processes for which they were designed, and to run them individually in the correct sequence (sometimes repeating execution when iterations are required) to correctly model the entire situation. Feng (2000) also recommends that modeling components be relegated to handling a specific hydrologic process, and that components be assembled together in a common environment with rules governing the communication between components. Pullar and Springer (2000) cite flexibility as an additional advantage attained by creating individual hydro components that can be assembled to perform work. Robbins and Phipps (1996) and Koussis et al (2003) used this approach to assemble integrated floodplain mapping applications with separate models for handling hydrologic and hydraulic processes.

Batelaan, Wang, and De Smedt (1996) developed WET-SPA (Water and Energy Transfers within and between the Soil, Plants, and Atmosphere), an object-oriented toolbox for modeling hydrologic processes. Depending on the temporal and spatial scale of the data, different models within the toolbox are chosen to suit the input.

An integration framework also enhances develop time for an integrated GIS and model application. Crow (2000) investigated several icon-driven development environments specifically designed for GIS and process components. Such environments use process flow diagrams to establish connections between data and processes, and to organize the processes into a meaningful sequence. Crow states that process flow diagrams allow a user to easily see what is going on in an integrated model, and allows the user to change inputs, parameters, or processes interactively.

Roberts and Moore (1998) state that not only is data structure important, but also the infrastructure required for browsing and downloading data. Roberts and Moore also stress the importance of metadata for data browsing and meaning. ESRI has addressed these issues through their ArcIMS internet software, which allows the browsing and downloading of GIS data as well as toolboxes containing geoprocessing work flows. Additionally, each ESRI toolbox contains a full metadata description for every tool and toolset in the toolbox.

Chapter 5 of this dissertation presents an integrated floodplain mapping model, assembled in the ESRI ArcGIS 9 ModelBuilder environment. ModelBuilder is an icon-based graphical development environment for linking individual processes and data sources to perform complex work. The model presented in Chapter 5 organizes GIS processes, HEC-HMS hydrologic processes, and HEC-RAS hydraulic processes in a sequence to create floodplain maps from rainfall data and channel characteristics. In Chapter 6, the ModelBuilder environment is used to create a water quality model. Using graphical workflow models provides a convenient environment to change data sources as well as hydrologic processors to conform to new situations.

2.1.3 Interface Data Models

Linking GIS and a simulation model requires a mapping of data between the two systems. Otherwise, conflicts or errors in data meaning could arise. GIS data models provide a solution to this problem. A GIS data model defines a structure for data storage within a GIS, including the categories of features stored, attributes of those features, and relationships between features. Arc Hydro facilitates the linking of GIS and hydrologic simulation models by defining features, attributes, and relationships useful to a wide variety of simulation models. However, Arc Hydro provides only the most common geospatial data needs of simulation models, such as stream network connectivity and watershed delineation.

Goodchild (1993) states that a GIS data model must fully support an environmental simulation model. This insures no ambiguity between data meaning in the two systems. Bian et al (1996) integrates SWAT (Soil Water Assessment Tool) and GIS using a separate graphical user interface and custom database to transfer data between the two. The data model for the application is based largely on, but not restricted to, data structures in SWAT.

The research presented in this dissertation introduces the concept of an Interface Data Model, which goes beyond Arc Hydro to provide direct compatibility between the GIS and a given hydrologic simulation model through common data meaning. Interface Data Models are described in Chapter 1.

2.1.4 Suitability of GIS to Model Time-Variant Processes

Correia et al (1998) observes that a GIS may be suitable for modeling processes that are not time-dependent, but recommends calling external simulation models when time series of data are used. Charnock, Hedges, and Elgy (1996) state that information transfer between models and GIS can be inefficient, especially when the GIS is the storehouse for model information, due to the large size of GIS files.

In Chapter 6 of this dissertation, a case study is presented in which the schematic processor calls hydrologic and hydraulic functions from the HEC's library of processing routines, LibHydro. Each function used the geodatabase to read and create time series data. This proved much more inefficient than calling the external HEC simulation models to perform the same work, which supports the conclusions presented by Correia et al (1998) and Charnock, Hedges, and Elgy (1996).

2.1.5 Regionalization

Streit and Kleeberg (1996) encountered the problem of transferring hydrologic information from small catchment scales to basin scales, and used generalization to address the issue. However, Streit and Kleeberg also recognized that generalization may introduce errors into the data. Schumann and Funke (1996) addressed this problem by creating components for rainfall-runoff modeling which utilize new scale-independent parameters. However, those applications were limited to a maximum basin size of roughly 10,000 km^2.

Abrahart et al (1996) developed an application which divided a study basin into sub-basins for use in MEDRUSH, a GIS combined with distributed process model designed to simulate hydrologic and vegetative processes important in studying desertification. Each sub-basin contains a set of flow-strips, in which hydrologic simulations occur. The total output from each flow-strip in a basin is converted to water and sediment output for that basin. Each basin feeds its total output to the stream network, where information is accumulated in the downstream direction. The application may operate on basins up to 5000 km^2 in area.

The research presented in this dissertation introduces a Raster-Network Regionalization Technique, in which large basins are subdivided into subregions suitable for the calculation of distributed watershed parameters. The subregional data is then merged to provide an integrated regional dataset attributed with the necessary hydrologic parameters. The technique has been successfully applied to the Rio Grande basin in Texas, which has a contributing area of roughly 470,000 km^2.

2.2 MODULAR MODELING SYSTEMS

2.2.1 MMS

The United States Geological Survey Modular Modeling System (MMS) provides a UNIX-based framework for simulating precipitation and run-off processes. MMS itself is not a model. Rather, it supports the connection of various models and data to form a complete simulation along with some additional visualization and component management functionality. A variety of simulation models can be linked together to model a system from several different perspectives. MMS also allows feedback mechanisms for iterations between models within the system. MMS consists of three main components: Pre-process, Model, and Post-process (USGS, 2002a).

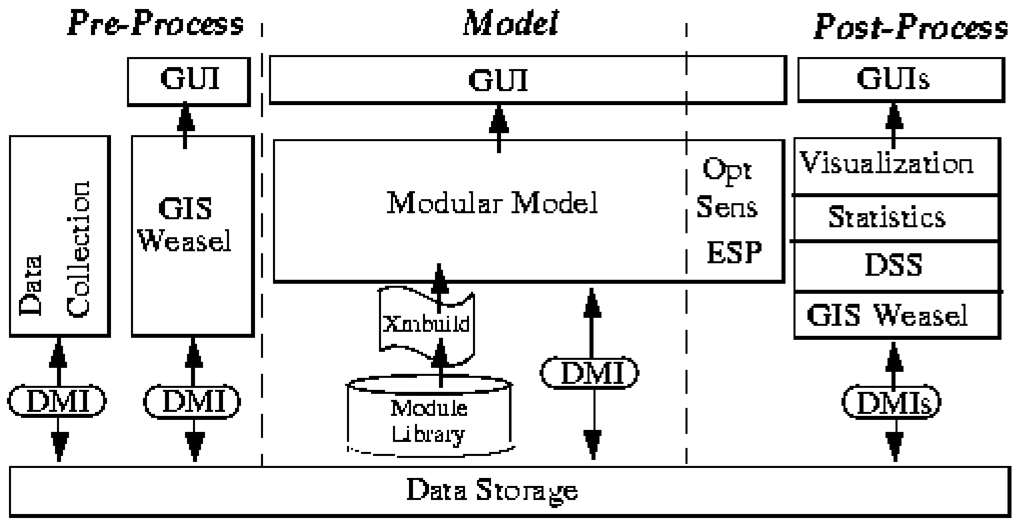


Figure 2.1 MMS Components (USGS, 2002a)

The Pre-process component handles data preparation for models. Functions in the Pre-process component perform analyses on the data, including spatial analysis using a GIS toolset. Once the data preparation process is completed, the data are written to spatial and temporal databases. These databases serve as the interface between Pre-process and Model components. The current structure for the time series database is an ASCII flat file. The Pre-process components must write to the databases using the correct format for data; otherwise, the Model components will not be able to interpret the data in a meaningful manner. This format is prescribed by MMS and resembles other text formats used for time series data by the USGS (USGS, 2002a).

The Model component contains all of the simulation models used by MMS. The models are selected from a module library, which contains various modules for simulating water, chemical, energy, and biological processes. The module library may contain more than one module to describe a single process. This gives the user the flexibility of selecting the most appropriate model given the conditions and data available for the simulation. The Model component reads in data from the spatial and temporal databases, runs through the models in the process flow set up by the user, and then sends output back to the databases for visualization. MMS provides the user with a graphical user interface for managing data preparation, model execution, and analyses of the results. Another interface allows the user to select modules and link them together to form an integrated model for describing a given system. In addition to standard model runs, MMS also allows for optimization runs and sensitivity analysis.

The Post-process component works with model outputs. This component contains statistical and graphical tools, as well as any user-developed tools for data analysis.

Data is converted from disk storage to a format that MMS components can interpret using a Data Management Interface (DMI). The DMI is compiled from C code, and is run each time parameters for model runs are to be extracted from disk.

One of the plusses of MMS is that the system supports the linking of any model into the system, provided that the model is encapsulated in a module that conforms to MMS rules. These rules specify a minimum of four functions that each module must implement: declare, initialize, run, and main. These functions handle the basic operation of a module, including the simulation run. Each module must also implement MMS library functions, which allow communication between different components of the system. The modules must be written in FORTRAN or C.

By following the MMS rules for communication, operation, and data storage (specially formatted ASCII files), any simulation model can be added to the MMS module library. However, it may be difficult, or even impossible, to wrap or alter legacy hydrologic models to conform to this standard. An additional consideration to using MMS is that the software is not designed to run in the Windows environment.

2.2.2 Open Modelling Interface and Environment

The Open Modelling Interface and Environment (OMI) is a framework for linking models covering different disciplines together in a coherent manner for the purpose of supporting integrated catchment management. The OMI is being developed by HarmonIT, a project funded by the European Commission to carry out mandates set forth by the Water Framework Directive. This directive recognizes the need for an integrated catchment approach to solving water resources problems, and aims to provide the tools required by engineers to carry out that approach. An integrated catchment approach not only examines a problem from a hydrological or hydraulic standpoint, but also from ecological, socio-economic, and other perspectives (HarmonIT, 2002).

The OMI is still in preliminary phases of development. However, an extensive literature and model review was conducted to learn from other efforts in the same arena. HarmonIT has identified six major issues for creating an integrated modeling framework (Hutchings, 2002):

- **Scale** – Models may operate at different temporal and spatial scales. Interpolation or extrapolation routines may solve temporal scale issues, while generalization or downscaling routines may solve spatial scale issues.
- **Dimensionality** – Models may operate in 0, 1, 2, or 3 dimensional space, with steady state or time varying conditions. While temporal dimensionality issues have been dealt with successfully in some applications, an elegant method of automating linkages across spatial dimensions has not been found through HarmonIT's research.
- **Parameter meaning** – Models tend to have their own terminology for naming parameters and features in the system. The capability to share or communicate data among models requires a way to understand parameter meanings. One way to solve this problem is with an extensive dictionary of acceptable parameter names and their definitions. Each model in the system would share its public data using the terminology from the dictionary. Another approach would be to create transformations (such as a style sheet) to convert one model's output into a form that can be used for another model's input.
- **Process control** – A robust mechanism for controlling the flow of work between models must be developed. This flow could involve purely sequential processing, where one model performs a complete simulation before providing input to the next model in the chain. The flow could also involve feedback iterations and looping, which is a much more

complicated case. It may be necessary to specify that certain models have priority for running first at a given time step. As this process control can become quite complicated, the framework developer must determine to what extent is the setup of process control exposed to the user.

- **Platform/language and communications issues** – Many models have been written in different programming languages and for different platforms (e.g. UNIX or PC). Some models take advantage of distributed computing over networks, while others do not. A truly generic integrated modeling framework should work in any of these environments with models written in any language.
- **Legacy models** – There are many powerful and proven hydrologic and hydraulic models in existence. However, no framework exists to integrate these models in an easy manner. An integrated modeling system should take advantage of the wealth of legacy models available, rather than require the user to develop new models that conform to framework requirements. Several methods exist to integrate legacy models into new frameworks. For example, wrappers could be written to serve as the bridge between a model and the integrated modeling framework. This wrapper would handle data exchange and possibly execution of the model. In some cases, models have to be completely rewritten to conform to the framework standard. However, cases exist in which no known method can be applied to a legacy model without an extreme amount of effort.

HarmonIT identifies data definition and the incorporation of legacy models as the key technical issues in developing an integrated modeling system (Hutchings, 2002).

2.3 Prototype Hydrologic Information Systems

2.3.1 EPA BASINS

EPA's Better Assessment Science Integrating Point and Nonpoint Sources (BASINS) combines a GIS (ArcView 3), modeling tools, and national watershed data in a functional system to support water quality-based assessment and analysis of pollution sources. BASINS is designed to support Total Maximum Daily Load studies by providing: (1) tools for examination of environmental information, (2) an integrated watershed and modeling framework, and (3) a system for evaluating point and nonpoint source management alternatives. BASINS is designed to work with ArcView GIS 3 (Olivera, 2002).

The data used in BASINS conforms to a particular data model, which defines the data structure and attributes of features. By following this structure, the tools in BASINS can recognize the data and extract any required information from the data.

The components of BASINS may be divided into five categories: (1) national environmental databases, (2) assessment tools, (3) utilities, (4) watershed characterization reports, and (5) simulation models and postprocessors (Environmental Protection Agency, 2001).

National Environmental Databases

Data for use in BASINS is available for download from the EPA web site by 8-digit Hydrologic Unit Code watersheds. These data cover the United States and include the following information:

- **Base Cartographic Data** – EPA regions, state and county boundaries, hydrologic unit boundaries, major roads, urbanized areas, populated place locations
- **Environmental Background Data** – National Hydrography Dataset, digital elevation models, STATSGO soils data, land use data
- **Environmental Monitoring Data** – National Sediment Inventory stations and database, water quality monitoring stations and data, bacteria monitoring stations, USGS gage stations, drinking water sites, weather stations and database, listing of fish and wildlife advisories, classified shellfish areas
- **Point Sources/Loading Data** – Permit Compliance System (PCS), Superfund national priority list, Resource Conservation and Recovery Information System (RCRIS), Industrial Facilities Discharge (IFD), Toxic Release Inventory (TRI), Minerals Availability System/Mineral Industry Location System (MAS/MILS)

These national databases provide the BASINS system with the necessary information for performing water quality analyses. BASINS is hard-coded to work with data from the EPA web site, which means that incorporating data from other sources is difficult. BASINS looks for data with a particular organization, structure, and format. The benefit of this approach is that the data works very well within the BASINS system. Although this approach discourages users from importing data from other sources, the BASINS databases are very comprehensive, and the EPA maintains the most current information online for download.

Assessment Tools

The assessment tools operate within the GIS. They automatically select features and generate informative charts given certain criteria. The criteria must follow a certain structure recognizable by BASINS, as do the data. The output from the assessment tools helps users identify watersheds or point locations that are critical in addressing water quality concerns. These tools do not perform operations that the user could not accomplish manually. However, they do automate many of those operations and allow for rapid visualization of problem areas. The following assessment tools are included with BASINS:

- **Target** – This tool identifies watersheds where a specific target concentration of a pollutant or water quality constituent is exceeded, or where discharges are permitted.
- **Assess** – This tool generates a map of monitoring stations classified according to a constituent concentration or permitted discharge load. The scope of this tool is limited to a single watershed.
- **Data Mining** – This tool defines a smaller water quality database based on a sub-set of selected stations.

Each of the above tools is listed in order of decreasing spatial domain. The Target tool works with the entire basin of interest. The Assess tool considers data in a specific watershed within the basin. The Data Mining tool extracts a subset of data from a given watershed for more detailed analyses.

Utilities

BASINS provides six utilities for management and general visualization of BASINS data. These utilities include:

- **BASINS Data Import** - Gives the user the ability to import additional data sets and prepares the data to work properly with BASINS functions and models
- **DEM Reclassification** – Modifies the legend of a digital elevation model (DEM) for display purposes
- **Land Use Reclassification** – Groups or renames land use categories to support analysis and modeling
- **Watershed Delineation** – Delineates watersheds from a digital elevation model
- **Lookup Table Query and View** – Provides quick access to relevant reference information and data products included with BASINS, such as map projection, agency codes, and water quality criteria
- **Water Quality Observation Data Management** - Accesses and manipulates water quality observation station information and data

Watershed Characterization Reports

BASINS provides tools for summarizing watershed information, from water quality aspects to topography, in order to prepare watershed characterization reports. These reports include maps and tables describing the condition of the study area, as well as information about point and nonpoint sources in the study area. The reports are useful as a decision-support aid, and may cover the following topics:

- Point Source Inventory
- Water Quality Summary
- Toxic Air Emission
- Land Use Distribution
- State Soil Characteristics
- Watershed Topography

Simulation Models and Postprocessors

Several water quality and watershed models are included in the BASINS system to simulate watershed processes and to predict the impacts of management scenarios. While each model was designed and may exist separately from the BASINS system, they have been integrated into BASINS so that the flow of data to and from models is managed through BASINS. The simulation components of BASINS include:

- **QUAL2E** – This water quality stream model is designed for 1-D, steady state conditions. The model requires information on channel networks and pollutant loadings from the BASINS database. Default values are provided for kinetic coefficients used by the model.

- **SWAT** – The Soil Water Assessment Tool is a time-variable watershed-scale model developed to predict the impact of land management practices on water, sediment, and agricultural chemical yields on large, complex watersheds with varying soils, land use, and management conditions over long periods of time. The model operates with 1-D, non-steady state conditions.
- **WinHSPF** – This time-variable watershed model simulates nonpoint and source runoff and pollutant loadings for a watershed and performs flow and water quality routing in reaches. The model operates with 1-D, non-steady state conditions.
- **GenScn** – This post processing and scenario analysis tool analyzes output from WinHSPF and SWAT.
- **WDMUtil** – This tool manages and creates watershed data management files (WDMs) that contain the meteorological data and other time series data used by WinHSPF.
- **PLOAD** – This GIS and spreadsheet tool calculates nonpoint sources of pollution in watersheds.

Each simulation model in BASINS, as well as the other BASINS components and databases, are integrated into the BASINS system, such that each part of the system can interact with other parts under the BASINS umbrella without conversion and workflow management required by the user. The result is a complete system for water quality modeling at the watershed scale with integrated data, tools, and simulation models.

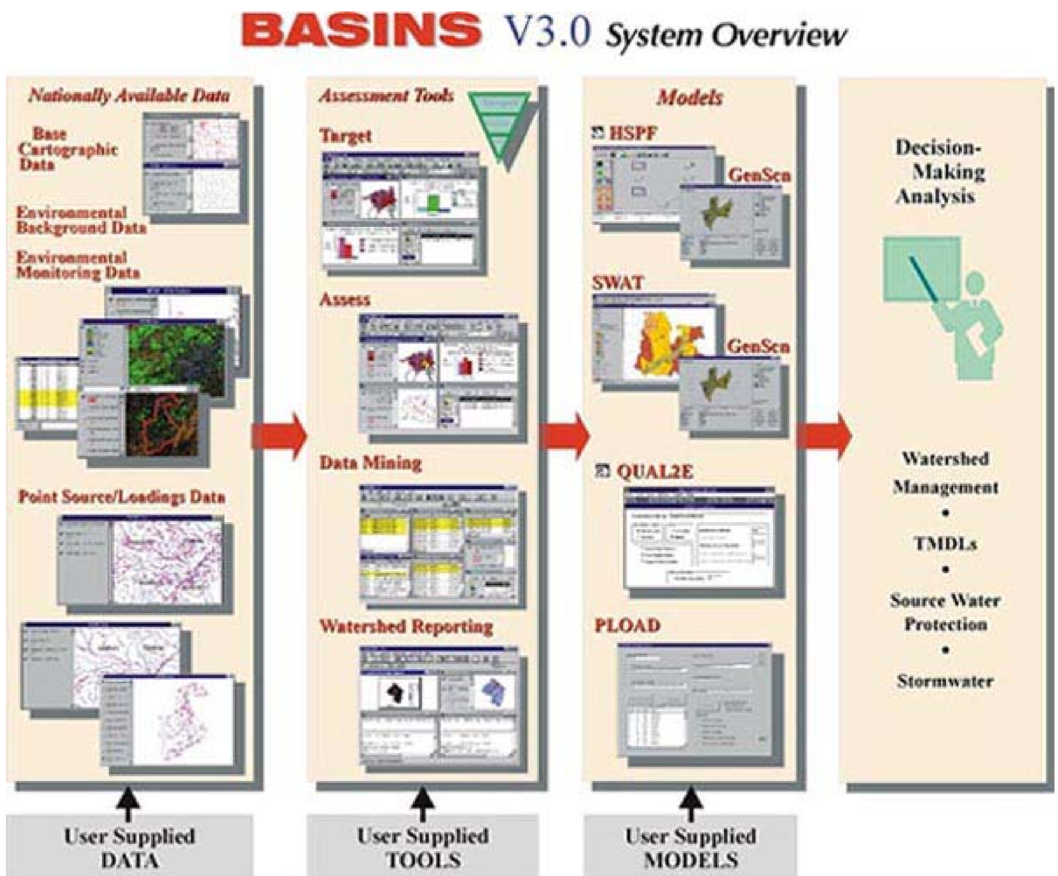


Figure 2.2 BASINS System Overview (Environmental Protection Agency, 2002)

The central interface for the BASINS working environment is through ArcView GIS 3. BASINS tools and models are linked to the system using Avenue, the customization programming language for ArcView. BASINS demonstrates how data and models can be linked in a hydrologic information system dealing specifically with water quality in streams at the watershed scale.

A harmonious system was created by focusing on a specific purpose and requiring a particular structure for the data in the system (Environmental Protection Agency, 2001).

2.3.2 HEC-HMS

The US Army Corps of Engineers Hydraulic Engineering Center's Hydrologic Modeling System (HMS) utilizes seven objects to route water through the landscape: Subbasin, Reach, Reservoir, Junction, Diversion, Source, and Sink. Each object is tagged with a unique identification code. Connectivity between objects is established by storing the identifier of the next downstream object for each object. The objects themselves are defined by the manner in which they conduct water through the landscape. The seven objects attempt to capture the essence of the major surface water features (HEC, 2001).

In HMS, each object receives an inflow time series, processes that time series based on object type and properties, and produces an outflow time series that is passed to the next downstream object. HMS assumes that no backwater effects occur, which means that there are no feedback loops between objects in the system. Thus, each object acts basically as a time series processor that is not concerned with the presence or nature of other objects in the system (HEC, 2001).

An HMS simulation is defined by three components: the Basin Model, the Meteorological Model, and the Control Specifications. The Basin Model contains a schematic consisting of any combination of the seven objects described above. The Basin Model stores information about the properties and connectivity of the objects in the schematic. The Meteorological Model contains time series information consisting of rainfall and evaporation data. These data are associated with rain gages that the user defines in the Meteorological Model. The Control Specifications component defines simulation properties such as duration and time step (HEC, 2001).

2.3.3 Water Availability Model

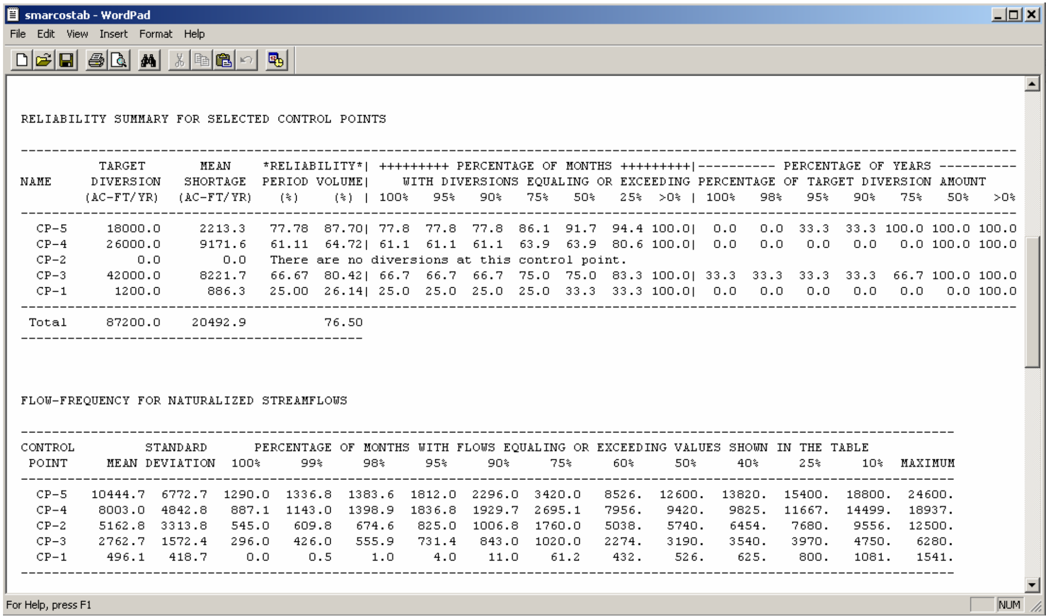
The State of Texas manages the supply and demand of surface water through a priority-based system of water rights. Water users must apply for permits and adhere to rules in the Texas Water Code regarding surface water usage. In 1997, the Texas legislature passed Senate Bill 1, mandating improved management and modeling capabilities for surface water resources in Texas. As part of the bill, the Texas Commission on Environmental Quality (then known as the Texas Natural Resource Conservation Commission) was given the task of developing water availability models for the major river basins in Texas. These models help water resources planners understand how much water was in the system, how that water is being allocated, and the impacts of additions or changes in water rights permits (Hudgens and Maidment, 1999).

The Water Rights Analysis Package (WRAP) was chosen by the Texas Commission on Environmental Quality (TCEQ) as the model to simulate water allocation. Hudgens and Maidment (1999) developed a set of ArcView GIS tools to calculate certain input parameters for the WRAP model based on geospatial data. The GIS tools were packaged as an ArcView 3 project file called WRAP 1117. WRAP and WRAP 1117 are described below.

Overview of WRAP

The Water Rights Analysis Package was developed at Texas A&M University under the supervision of Dr. Ralph Wurbs (Wurbs, 2001). WRAP supports assessment of water availability and reliability for water rights in a priority-based water allocation system. Water rights can represent anything from a reservoir operating system, a municipal water supply, or irrigation withdrawals. WRAP uses a record (about 50 years in typical applications) of monthly naturalized streamflows (naturalized streamflows are those that would exist naturally without the influence of man or man-made structures and diversions), reservoir evaporation rates, basin characteristics (such as curve number and drainage area), and mean annual precipitation rates to simulate the hydrology for a basin. WRAP uses water right priorities, rules, and target relationships to allocate water during the simulation period using a monthly time step. Simulation results include such information as reservoir storage levels and reliability indices for meeting water use requirements (Wurbs, 2001).

WRAP consists of a set of Fortran programs for preparing simulation inputs, running a simulation, and summarizing simulation outputs. The default WRAP user interface is the command prompt. Each component of WRAP is accessed by calling the appropriate executable. WRAP-HYD is a pre-processing component for preparing naturalized streamflows and reservoir net evaporation-precipitation rates. WRAP-SIM performs the allocation simulation. TABLES is a post-processing component for organizing and summarizing simulation outputs (Wurbs, 2001).



RELIABILITY SUMMARY FOR SELECTED CONTROL POINTS

NAME	TARGET DIVERSION (AC-FT/YR)	MEAN SHORTAGE (AC-FT/YR)	*RELIABILITY* PERIOD (%)	*RELIABILITY* VOLUME (%)	+++++++++ PERCENTAGE OF MONTHS +++++++++ WITH DIVERSIONS EQUALING OR EXCEEDING PERCENTAGE OF TARGET DIVERSION AMOUNT 100%	95%	90%	75%	50%	25%	>0%	---------- PERCENTAGE OF YEARS ---------- 100%	98%	95%	90%	75%	50%	>0%
CP-5	18000.0	2213.3	77.78	87.70	77.8	77.8	77.8	86.1	91.7	94.4	100.0	0.0	0.0	33.3	33.3	100.0	100.0	100.0
CP-4	26000.0	9171.6	61.11	64.72	61.1	61.1	61.1	63.9	63.9	80.6	100.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0
CP-2	0.0	0.0	There are no diversions at this control point.															
CP-3	42000.0	8221.7	66.67	80.42	66.7	66.7	66.7	75.0	75.0	83.3	100.0	33.3	33.3	33.3	33.3	66.7	100.0	100.0
CP-1	1200.0	886.3	25.00	26.14	25.0	25.0	25.0	25.0	33.3	33.3	100.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
Total	87200.0	20492.9		76.50														

FLOW-FREQUENCY FOR NATURALIZED STREAMFLOWS

CONTROL POINT	MEAN	STANDARD DEVIATION	PERCENTAGE OF MONTHS WITH FLOWS EQUALING OR EXCEEDING VALUES SHOWN IN THE TABLE 100%	99%	98%	95%	90%	75%	60%	50%	40%	25%	10%	MAXIMUM
CP-5	10444.7	6772.7	1290.0	1336.8	1383.6	1812.0	2296.0	3420.0	8526.	12600.	13820.	15400.	18800.	24600.
CP-4	8003.0	4842.8	887.1	1143.0	1398.9	1836.8	1929.7	2695.1	7956.	9420.	9825.	11667.	14499.	18937.
CP-2	5162.8	3313.8	545.0	609.8	674.6	825.0	1006.8	1760.0	5038.	5740.	6454.	7680.	9556.	12500.
CP-3	2762.7	1572.4	296.0	426.0	555.9	731.4	843.0	1020.0	2274.	3190.	3540.	3970.	4750.	6280.
CP-1	496.1	418.7	0.0	0.5	1.0	4.0	11.0	61.2	432.	526.	625.	800.	1081.	1541.

Figure 2.3 Sample WRAP Output - Reliability Summary

The inputs and outputs of WRAP are all specially formatted text files. These text files consist of records with information about each record. A record occurs on a single line in the input file. Information about a record occurs at specific columns along the line for that record. Because WRAP is Fortran-based, the parameter entries for each record must occur at the correct column position in the text file, with ‘spaces’ filling in the white space between columns. For example, the curve number for a given watershed parameter record must occur in columns 17-24 (Wurbs, 2001). An input file may contain thousands of records of various types. A simple example with flow distribution and watershed parameter records is given below.

```
**  Example 4 -  WRAP-SIM Input File Exam4.DIS
**  Example 4 from Appendix I of Manual
**
**  Flow Distribution Information
**
FD  CP-4    CP-5       2    CP-2    CP-3
FD  CP-1    CP-2
**
WP  CP-1      225      74      31
WP  CP-2      398      69      31
WP  CP-3      194
WP  CP-4      650
WP  CP-5      715
```

Figure 2.4 Sample WRAP Input File (Wurbs, 2001)

For a given simulation, WRAP reads a set of input files that each have the same root name, with different extensions. For example, the sample file in the figure above could reside as sample.DIS, where 'sample' is the root name of the simulation. In a typical simulation, the following input files are used:

- **root.DAT** – Basic input data file, sans hydrology information provided in the files below
- **root.INF** – records of naturalized streamflows
- **root.DIS** – flow distribution and watershed parameter records
- **root.EVA** – records with net evaporation-precipitation rates

WRAP 1117 ArcView Project

To facilitate the calculation of control point and watershed parameters for WRAP, Hudgens and Maidment (1999) created a set of tools encapsulated in an ArcView 3 project file called WRAP 1117. This file contained Avenue scripts linked to a series of menus and menu items, which were used to prepare the stream network and a digital elevation model so that average curve number, mean annual precipitation, drainage area, and connectivity could be determined for the watershed draining to each control point on the stream network. The first three parameters are calculated using raster analyses. The procedure involves burning streams into a DEM, filling sinks, calculating a flow direction grid, and then calculating flow accumulation grids. The drainage area value over a given point is simply the number of cells accumulated to that point, times the area of each cell. A weighted flow accumulation is used to determine the average curve number and mean annual precipitation over a given point. Thus, those three parameters are calculated for every single cell over the extent of the DEM, and the value for a given control point is found by picking up the grid cell value for the cell underneath that control point. The connectivity of control points is determined by identifying the next downstream control point in the stream network. Once the parameters are calculated for all control points, the results are then input to a WRAP model (Hudgens and Maidment, 1999).

While WRAP 1117 provides useful tools for performing watershed analyses with small watersheds, the tools encounter scalability problems when applied to large river basins. In basins large enough to cover more than 100 million cells in a raster dataset, processing demands may exceed a computer's processing capabilities (Figurski, 2001).

2.4 CONCLUSIONS FROM LITERATURE REVIEW

Applications such as WRAP 1117 and EPA BASINS show the value in integrating GIS and hydrologic simulation models. BASINS and HMS reveal that a given hydrologic simulation model or application requires a specific data model to support that application. Likewise, MMS shows that a specific set of rules regarding the flow of information between models is necessary for successful integration of simulation models.

The applications surveyed in this literature review also reveal gaps in the state of knowledge regarding GIS and model integration. While a GIS is useful as a preprocessor for model inputs, the application of WRAP 1117 has shown that a methodology is needed to facilitate raster processing of large basins for hydrologic parameter calculation. Research performed by the Open Modelling Interface and Environment effort suggests that data definition and the incorporation of legacy models present tough challenges for the integration of simulation models. The difficulty in importing and using data that does not conform to the BASINS data structure into a BASINS database, as well as the difficulty in incorporating legacy models into MMS, supports the OMI's conclusions.

This research shows how to use new functionality made available in ArcGIS 8 and 9 to address these issues.

- The Raster-Network Regionalization Technique uses the Arc Hydro and WRAP Hydro tools for hydrologic processing of large regions.

- Interface Data Models, which are used to communicate between simulation models and Arc Hydro data in a GIS, serve as a bridge between different simulation models by using Arc Hydro as a common ground for communication. Information exchange points provide a common location for exchanging data between models through a GIS. With these concepts, Arc Hydro and ModelBuilder are used to integrate HEC-HMS and HEC-RAS into a single workflow model to calculate a floodplain map from a rainfall map.
- This research initiates a study of attaching hydrologic behavior to GIS features using a schematic processor. Arc Hydro schematic networks and a schematic processor are used to achieve an even tighter integration between GIS data and mechanisms for simulating hydrologic behavior.

These techniques for geographic integration are presented in Chapters 4, 5, and 6.

Chapter 3 Technology Review

This chapter provides a review of programming practices and technology utilized by this research.

3.1 Programming and Design Techniques

Computer software and programming practices play a strong role in the development of a hydrologic information system. This section provides an introduction to object-oriented programming and COM, software design principles, and user interface design.

3.1.1 Software Design Principles

The philosophies behind software construction have almost become like religions in the programming world, from top-down approaches to bottom-up approaches. However, Meyer (1997) does identify some useful principles in software construction that may be applied to almost any situation. These principles are *robustness*, *extensibility*, *reusability*, and *compatibility*.

Robustness measures the ability of a program to respond gracefully to circumstances that fall outside of the original design constraints. These circumstances could be caused by unforeseen errors, or by applying the software to new conditions. For instance, a user may try to apply a rainfall/runoff model designed for mountainous slopes to an area with flat terrain. A robust model, while perhaps not calculating correct runoff estimates, will still complete execution without catastrophic failure (i.e., crashing) (Meyer, 1997).

Extensibility measures the ease with which modifications can be made to software to address new needs or specifications. Design needs and application scenarios change over time, as do technology and programming practices. A non-extensible piece of software may work fine for its time, but may fade into obscurity when users realize "you can't teach an old model new tricks." Today engineers are confronted with this problem when trying to extend valuable legacy models for use in current applications. A simple, decentralized and modular design is key to an extensible design (Meyer, 1997).

Reusability measures the ease with which software components may be reused for other applications. One piece of software may provide functionality that is useful in other software applications, such as creating a chart from time series values. By incorporating components from existing programs, developers save much time and effort on development. Also, those components have already been tested and proven through previous applications (Meyer, 1997). Extracting useful components from software packages is one way in which software designers are dealing with legacy models.

Compatibility measures the ease with which a piece of software may be combined with other software (Meyer, 1997). One way to improve compatibility is to implement a COM-compliant design, where COM is a binary specification standard that allows software to communicate with or incorporate components from other software that follows the COM standard. Compatibility is a key issue in assembling a hydrologic information system, which integrates a variety of components.

Another important aspect of software construction is the user interface design. An otherwise well-designed piece of software will prove useless without an effective means of communicating with the user. As software systems become more complex (especially in the case of a hydrologic information system), providing a friendly yet powerful user interface becomes more and more critical to successful software implementation. Hartley (1998) provides some useful guidelines to user interface design. Among these, a design should be simple and intuitive, so that the user does not get lost in a mass of buttons, menus, windows, with seemingly no coherent logical design. The user should also have a sense of control over the application. The application should perform the tasks desired by the user without a fuss. Consistency, feedback, and general aesthetics are also important issues. Many current applications mimic the standard Windows menus (e.g. File, Edit, Help) so that new users start off with a feeling of familiarity with the interface (Hartley, 1998).

3.1.2 Object-oriented Programming and COM

Object-oriented programming represents real-world concepts using constructs called objects. Objects possess properties, which describe the object, and methods, which consist of tasks that the object performs (Rumbaugh et al., 1991). For example, a river object could be described by geometric properties, and it could have a method that calculates the volume of water in the river based on those properties and a surface water elevation. The latest GIS software from ESRI makes use of object-oriented programming both in its object library exposed for customization, and in the actual representation of vector features.

The Component Object Model (COM) is a binary specification standard that enables software that follows the COM standard to communicate with other COM-compliant software by accessing exposed object libraries. The object library for a given software contains objects that may be accessed (and whose properties and methods may be called) by other programs. As long as the COM standard is followed, even applications written in different programming languages may interact. An example of COM in practice can be seen with Microsoft's Word and Excel programs. The user can copy and paste a chart from Excel into Word, and then even edit the chart in Word. Behind the scenes, Word is accessing the Excel object library to figure out what to do with the chart. By incorporating COM into software design, that software is stepping into a circle with all the other COM-compliant programs, thereby greatly enhancing its potential for extended functionality, software integration, and user-acceptance. One of the reasons why ArcGIS was chosen as the GIS software for this research is because of its COM-compliant design.

3.1.3 Scripts, Executables, and DLLs

Once a software application has been designed, it may be programmed and distributed as a script, executable, or dynamic linked library (DLL). The choice of which medium to use depends on the scope of the application and how the application was designed to operate.

A *script* consists of programming code stored as text (Alintex, 2003). The code within a script is written to conform to the rules of a script compiler, which interprets and carries out commands in the script. Each line in a script is executed, until the script is finished. A script is typically the least powerful of the three mediums listed above, but is very useful for producing code to perform relatively simple tasks without investing a great deal of development time. Scripts are written in languages such as Python, Java, and Visual Basic Scripting.

Executables are software applications packaged as a single file that is run by the user. Executables control their own lifetime. In other words, the file keeps running until some event triggers the executable to close. The executable typically responds to perform work as directed by the user.

DLLs are libraries of objects or procedures which are accessed by a calling application. A DLL does not control its own lifetime; it does not run on its own. Rather, an application (such as an executable) calls a DLL, accessing useful functions or objects from the DLL. Thus, an executable uses a DLL to provide functionality in addition to that provided by the executable alone. While a DLL is typically called by an executable, DLLs may also be called from scripts.

Scripts, *executables*, and *DLLs* may be accessed from other scripts, executables, or DLLs. All three of these entities are used in the execution of models and the linking of models with geospatial data in this research, with each entity having advantages and limitations. While scripts provide quick solutions to simple problems, they quickly lose favor as the task at hand becomes more complex. Executables are required when an external application is required. However, packaging related components of an executable into DLLs allows for components to be upgraded or replaced without having to recompile and replace the entire executable. DLLs also provide an increased level of process control over executables, since the calling application controls the lifetime of objects in a DLL.

Category	Advantages	Limitations
Script	Quick solutions to simple tasks	Inappropriate for complex tasks
Executable	Can run independently of other applications	Process control not as secure as DLLs
Dynamic Linked Library	Modularity in design promotes robustness and extensibility	Cannot run independently of other applications

Table 3.1 Advantages and Limitations of Scripts, Executables, and DLLs

3.2 ArcGIS Software

Geographic Information Systems (GIS) provide the user with powerful mapping and spatial analysis tools. With the ability to efficiently process data to produce input parameters for models, and visualization capabilities for displaying model outputs (such as flood inundation zones), a GIS can be a vital component of a hydrologic information system. This research utilizes the latest GIS software from ESRI, the world's leading producer of GIS software. ESRI's GIS software package is called ArcGIS, and is designed to work on a personal desktop computer or in a distributed network environment. ArcGIS incorporates some powerful technologies new to a GIS, such as RDBMS. RDBMS (relational database management systems) supports the creation of relationships between tables in a database. ArcGIS is the first GIS software released by ESRI with a COM-compliant design. Through COM, ArcGIS can now communicate with other COM-compliant software, such as Word, Excel, and Internet Explorer, by utilizing public components (those that can be accessed by other applications) from each software's object library.

An understanding of ArcGIS is important in understanding the ArcGIS Hydro data model, which is built to work with ArcGIS and plays a key role in developing a hydrologic information system using GIS. The researched presented in this dissertation was implemented using two versions of ArcGIS: ArcGIS 8 and ArcGIS 9 beta. ArcGIS 9 will be the next release of ArcGIS, and possesses several key enhancements over ArcGIS 8. A beta version of ArcGIS 9 was available for this research, and provided functionality which made the research possible. This section presents an overview of ArcGIS 8, followed by a description of the components of ArcGIS 9 utilized by this research.

3.2.1 User Interface

ArcGIS is composed of three major components: ArcCatalog, ArcMap, and ArcToolbox. ArcCatalog resembles Windows Explorer in look and feel, its purpose being the creation, cataloging, and maintenance of GIS files. With ArcCatalog, the user can browse and manage data, preview data (either through a spatial display or by showing attributes in a table), and access metadata. ArcGIS supports several formats and standards for storing metadata, including FGDC, ISO, and XML. ArcCatalog also establishes relationships between features, and builds geometric networks (which are used to establish a network topology between point and line features.)

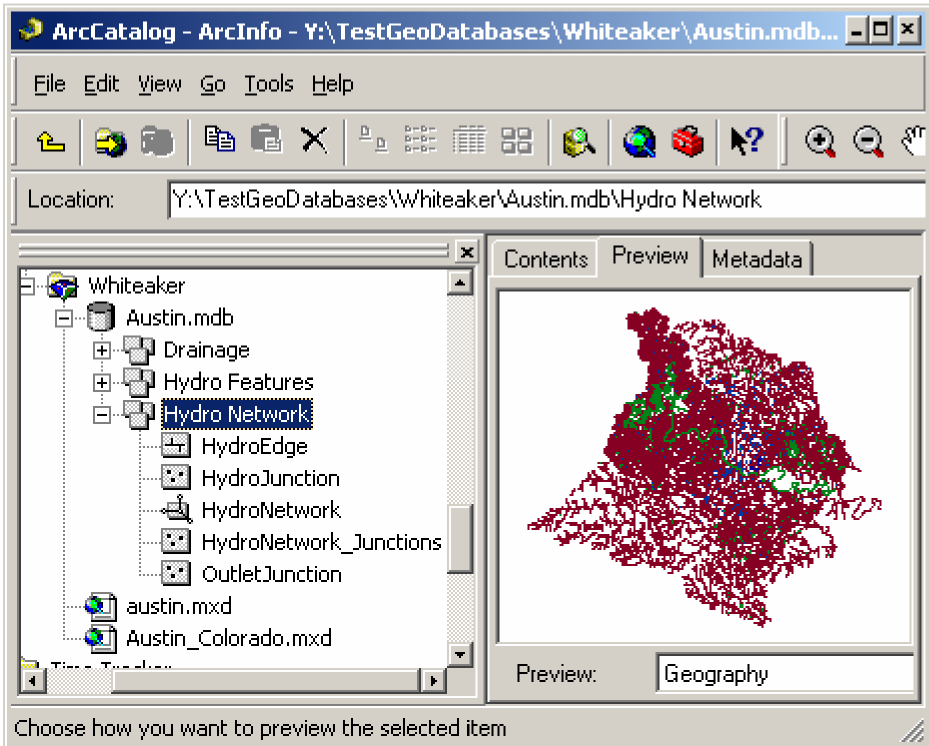


Figure 3.1 Geodatabase in ArcCatalog Graphical User Interface

ArcMap is used for display and analysis of GIS data. It contains tools for zooming, panning, and making printable maps, as well as special editing and network tracing tools. The ArcMap user interface may be customized by creating custom tools, and then associating those tools with code that performs a particular task. Visual Basic for Applications (VBA) is the primary customization language for ArcGIS.

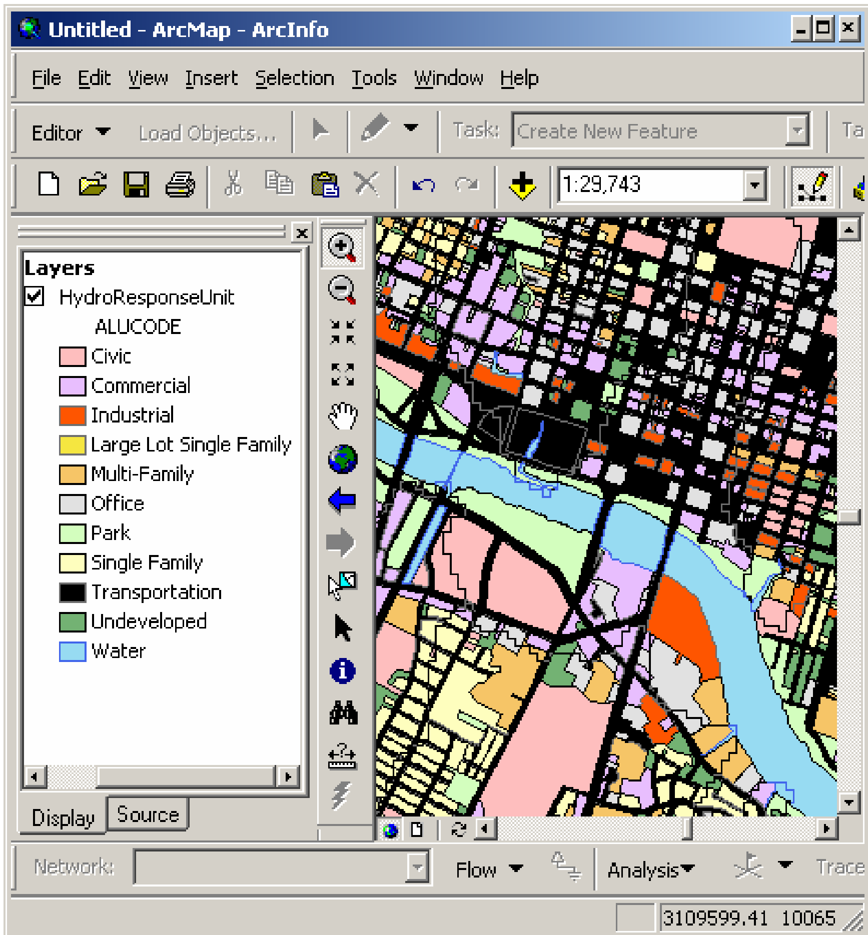


Figure 3.2 Land Use Data for Austin, TX, Shown in ArcMap

ArcToolbox contains utilities that operate on GIS data, including projection utilities and data conversion tools. ArcToolbox tools typically take some input (an input database and/or parameters), and perform a processing task to produce a new output dataset. In ArcGIS version 8, ArcToolbox is a separate application. In ArcGIS version 9, the toolbox is a dockable window in both ArcCatalog and ArcMap.

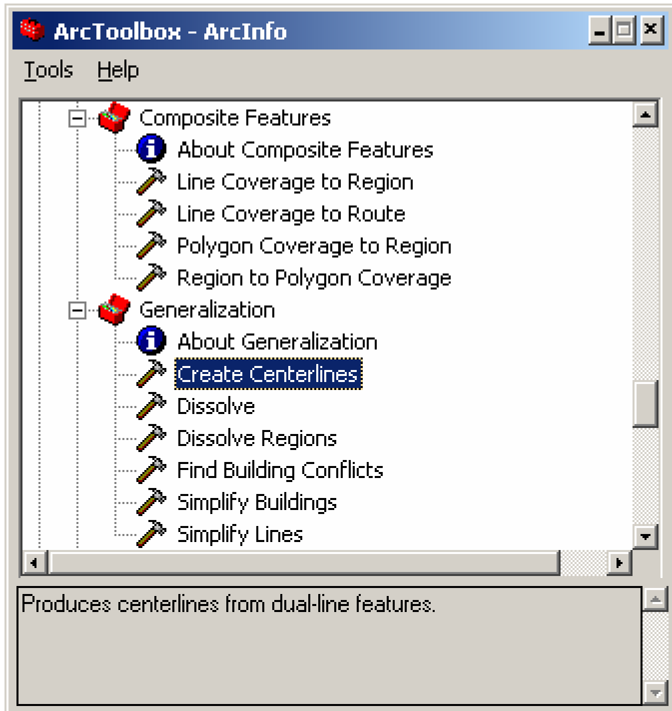


Figure 3.3 ArcToolbox Graphical User Interface

3.2.2 Geodatabase Structure

ArcGIS utilizes a relational database management systems (RDBMS) structure to store data. A key benefit of RDBMS technology is the ability to store data in a blob field. This is a special type of field that can store any type of data, from traditional data types (such as integers and strings) to files and images. ArcGIS uses the blob field to store the geometry of features. Thus, a given Feature Class is described by its attributes as fields in a table, along with its shape as another field in the same table. This is different from previous GIS data structures, such as shapefiles, which store a features attributes in a separate file from its geometry.

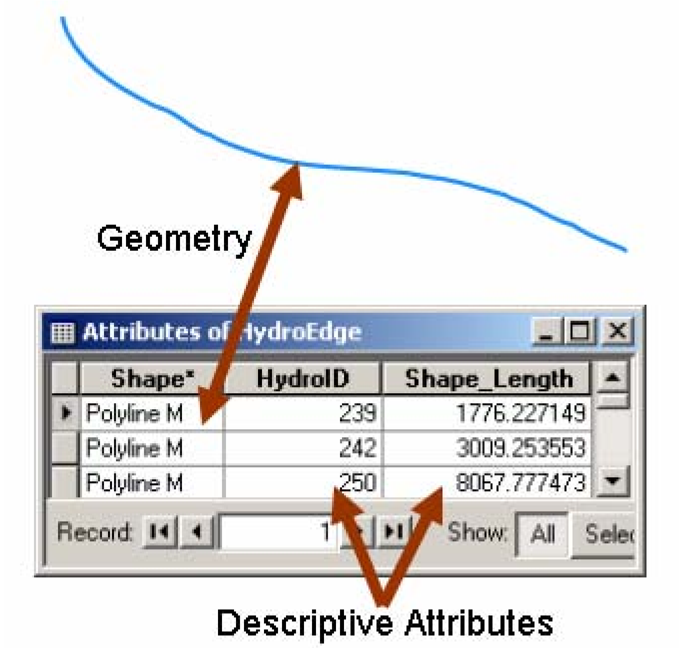


Figure 3.4 Feature Class Table Structure

In ArcGIS, spatial data are stored in a geodatabase, which is just a database (such as a Microsoft Access database) structured for use with ArcGIS. In ArcGIS 9, a geodatabase may contain vector data, raster data, relationships, networks, and custom tools. Raster data refers to images or grids, such as digital elevation models or satellite imagery. Vector data consist of points, lines, and polygons. The following terminology is used to describe categories of vector data:

- **Feature** – an individual spatial entity, represented by one row in a geodatabase table
- **Point** – point feature
- **Polyline** – line feature

- **Polygon** – polygon feature
- **Geometric Network** – set of points and lines participating in a topologically connected network, upon which connectivity, flow direction, and tracing tasks may be performed
- **Simple Junction** – basic node in a geometric network
- **Generic Junction** – simple junction automatically created at the endpoint of an edge or where two edges meet in a geometric network
- **Complex Junction** – node with complex behavior in a geometric network
- **Simple Edge** – basic linear feature in a geometric network, bounded by two junctions with no junctions along the interior of the line (interior junctions); comprised of a one polyline geometry with one set of attributes
- **Complex Edge** – line that may contain interior junctions in a geometric network; comprised of one or more polyline geometries with one set of attributes
- **Simple Feature** – a point, line, or polygon not participating in a geometric network
- **Network Feature** – points or lines participating in a geometric network
- **Feature Class** – a collection of features of the same type, existing as a table in the geodatabase
- **Feature Dataset** – container for feature classes that defines the coordinate system for the feature classes; may also contain geometric networks and relationships

- **Geodatabase** – container for a set of spatial data and tools; may include feature datasets, geometric networks, relationships, standalone feature classes and tables, tools, and raster data
- **Relationship** – defines a link established through key fields between two feature classes

3.2.3 GeoProcessing in ArcGIS 9

ArcGIS 9 provides a new geoprocessing framework that allows geoprocessing tasks to be chained together in *workflow models*. These tasks may be arranged from the standard tools in ArcGIS 9, as well as custom tools created by the user and stored in custom toolboxes. Custom tools may be generated from developer's code, scripts, or GIS models. The terminology here may seem confusing, as a chain of geoprocessing tasks in the GIS is called a workflow model. This is not to be confused with a simulation model or a data model, as described in Chapter 1.

Creating tools using developer's code requires implementing certain geoprocessing interfaces in a language that supports COM, such as C++. While this approach is geared towards hardcore programmers, scripting provides a simple alternative to the user with less programming knowledge. Geoprocessing scripts are written using scripting languages such as Python, Java, or Visual Basic Scripting. The scripts simply set inputs and outputs and call geoprocessing tasks. The maintenance tasks of locating data, preparing workspaces, and prompting the user for necessary inputs are handled by the ArcGIS geoprocessor. Thus, by relying on the geoprocessor to handle many of the basic input/output operations, a script may accomplish a task in 10 lines of code, a task for which a tool made from developer's code may require 1000 lines of code.

ArcGIS 9 provides an environment called ModelBuilder for assembling geoprocessing tasks. A geoprocessing task typically takes one or more inputs, performs an operation on those inputs, and produces a derived dataset. A set of inputs, a geoprocessing task, and outputs make up a process. In ModelBuilder, a chain of processes makes up a model. The outputs from one process become the inputs to the next process in the chain.

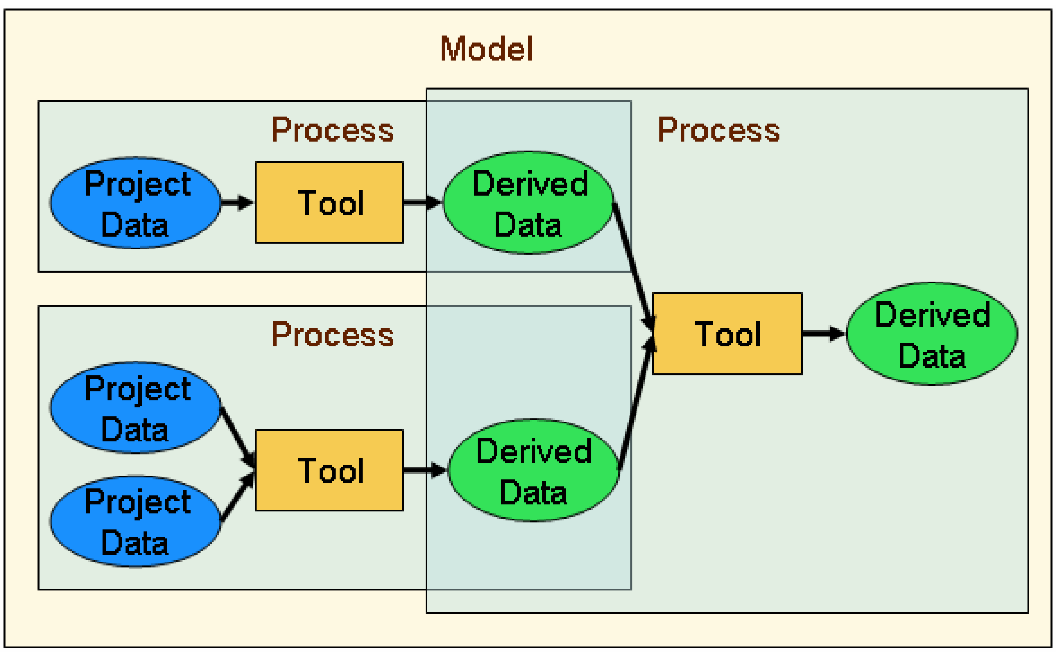


Figure 3.5 Basic Model Structure in ModelBuilder

To construct a model, a tool is selected from ArcToolbox, and then dragged onto a ModelBuilder diagram. A tool may consist of the standard ArcGIS tools, a custom script, or another model created by ModelBuilder. Each type of tool is differentiated by the icon on the tool in the ModelBuilder window. Once inputs, outputs, and any necessary parameters have been defined for the model, it may be run or saved in a toolbox.

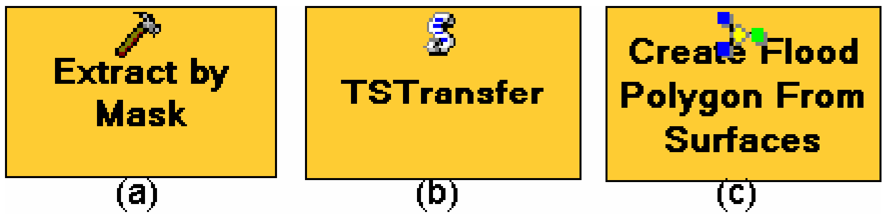


Figure 3.6 (a) Standard ArcGIS Tool, (b) Script Tool, and (c) Model Tool indicated by hammer, scroll, and flowchart symbol, respectively

While standard ArcGIS tools are suitable for many applications, sometimes more specialized or advanced functionality is required. The user may create this functionality by programming a script tool. Once a workflow model has been created, it may be reused and inserted into other models. Thus, the user has a variety of options for creating and assembling a workflow model.

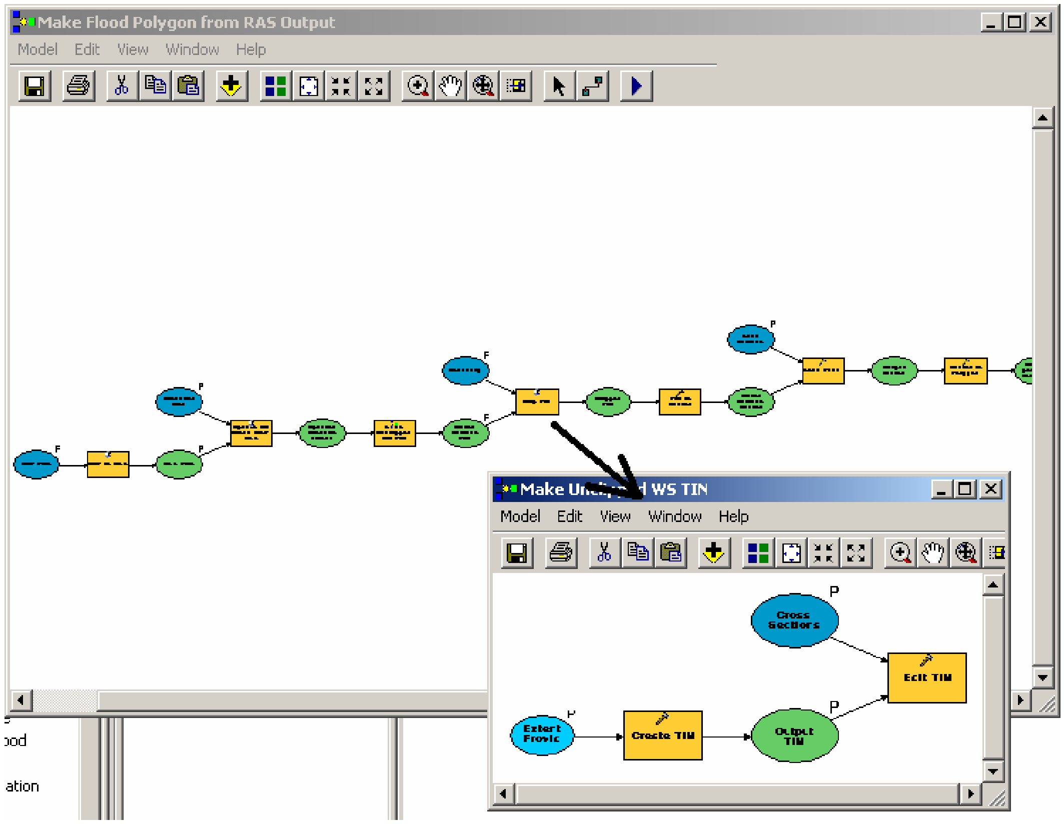


Figure 3.7 Complex Model in ModelBuilder: A model may contain tools, scripts, or other models

When executing a model, the geoprocessor reads and validates the model, and then automatically constructs an input form prompting the user for the model's necessary inputs. This form is XML-based, and has the same look and feel for every ArcToolbox tool (even custom tools that the user creates). When developing new functionality in ArcGIS, it is advantageous to expose that functionality as an ArcToolbox tool when appropriate. By letting the geoprocessor handle inputs, outputs, and data management, the development effort can be focused more on the actual functionality of the tool.

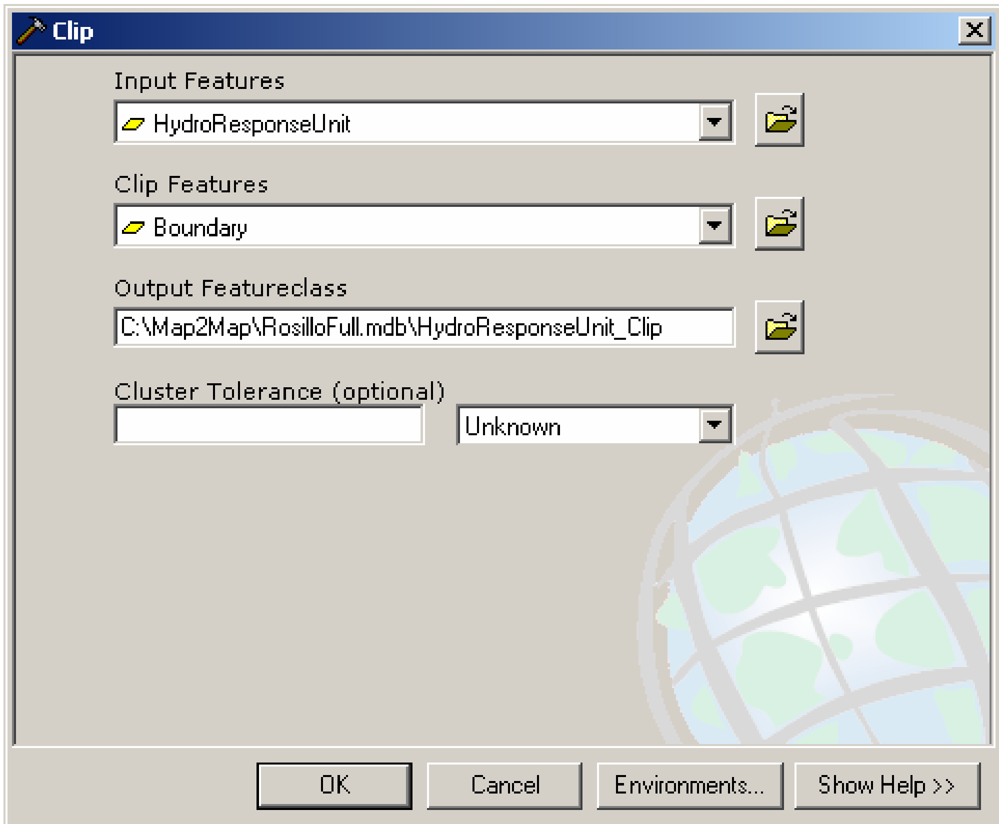


Figure 3.8 A Typical Geoprocessing Tool Interface

ModelBuilder gives the ArcGIS user the ability to perform complex geoprocessing tasks with little or no programming. By implementing custom functionality as tools, a user can reuse the tool by dragging it onto a new model diagram. No reprogramming of the tool is required, unless modified functionality is desired.

3.3 ARCGIS HYDRO DATA MODEL

The ArcGIS Hydro data model (Arc Hydro) is a water resources data model that uses GIS to capture the essence of surface water features. The goal of

Arc Hydro is to represent data in a manner that supports hydrologic simulation modeling (Davis, 1999). Arc Hydro defines a set of water resources feature classes, such as watersheds, cross sections, and monitoring points. Arc Hydro defines time series classes to handle time series data. In addition to prescribing an attribute structure and organization, Arc Hydro also defines relationships between features, so that a watershed may know which point represents its outlet, or a monitoring point may be aware of all time series records measured at that location (Maidment, 2002). An Arc Hydro toolset was developed along with Arc Hydro. This toolset creates Arc Hydro data, populates the attributes of Arc Hydro, performs operations using the Arc Hydro data structure, and supports visualization of time series data. Arc Hydro and the Arc Hydro tools were designed to work in the ArcGIS environment. However, the data structure of Arc Hydro could be applied to any GIS environment.

Arc Hydro supports hydrologic simulation modeling by establishing connectivity between hydrologic features in the landscape. This connectivity can be used to direct the flow of water between features in a simulation model. The Arc Hydro toolset also calculates certain attributes that may be useful in simulation models, either through attribute accumulation routines through relationships or network associations, or by direct calculation of parameters such as the length from a point on the network to the outlet of the river system. In essence, Arc Hydro is a data support system for hydrologic simulation. Arc Hydro serves another useful purpose as a standard for storing water resources data. This enhances the sharing of data among different organizations. Efforts have already been carried out to incorporate Arc Hydro into national applications such as the National Hydrography Dataset (Hickman, 2002).

3.3.1 Data Model Design

The ArcGIS Hydro data model was designed in the Unified Modeling Language, or UML. UML is a graphical standard for writing software blueprints (Booch, Rumbaugh, and Jacobson, 1999). Once software components have been designed visually on a UML diagram, a software engineering tool such as Visio 2000 may be used to convert the blueprint into a form which supports the actual creation of the components through programming code. While Arc Hydro features could support custom behavior through code generation with C++, currently no custom behaviors are implemented in Arc Hydro. Arc Hydro is composed of five major packages, with four packages representing geospatial Feature Datasets that will be included in the geodatabase, and the fifth package representing time series data. The packages are Hydrography, Network, Drainage, Channel, and Time Series. A summary of the Arc Hydro classes is given below. For a more complete description of Arc Hydro, refer to Maidment (2002).

3.3.2 Hydrography

The Hydrography feature dataset stores map-related features, and was designed with the structure of existing national datasets in mind. Twelve Hydrography feature classes are used to represent the cartographic features of the landscape.

- **HydroPoint** – A generic point feature on a map

- **Bridge** – A transportation structure spanning a river
- **Dam** – A structure that retains water to form a reservoir
- **Structure** – A water structure other than a bridge or dam, such as a waterfall
- **UserPoint** – A user-defined point location
- **WaterDischarge** – A point where water is discharged into a body of water
- **WaterWithdrawal** – A point where water is withdrawn from a body of water
- **HydroResponseUnit** – A uniform unit of the landscape describing hydrologic properties important in the vertical transfer of water between the atmosphere, surface, and subsurface
- **MonitoringPoint** – A point where water quantity or quality measurements are taken
- **HydroLine** – A linear hydrographic feature on the map, not represented by a HydroEdge
- **HydroArea** – An areal hydrographic feature on the map, not represented by a Waterbody
- **Waterbody** – An areal water feature in the landscape

A relationship ties a Waterbody to the HydroJunction that serves as its outlet. Relationships also connect a MonitoringPoint to its TimeSeries data, and to the HydroJunction that represents the MonitoringPoint on the stream network.

3.3.3 Network

The Network feature dataset contains the HydroNetwork used for tracing the flow of water through a stream network, as well as feature classes providing a schematic representation of topological connectivity and object classes used for defining events along linear features. The HydroNetwork is an ArcGIS geometric network representing the stream network, which supports network traces such as those available on the ArcMap Utility Network Analyst Toolbar. The HydroNetwork includes HydroEdge, HydroJunction, and HydroNetwork_Junctions feature classes. Connectivity between features in the HydroNetwork is based on geometry. For example, if two HydroEdges share an end point, then those features are connected in the network. The HydroNetwork closely resembles the cartographic representation of the stream network in shape, although small differences occur where the HydroNetwork has been adjusted to insure correct network topology.

An Arc Hydro schematic network provides a schematic representation of the connectivity between hydrologic features. Features in a schematic network are represented as nodes using the SchematicNode feature class, while links between features are represented with straight lines between SchematicNodes using the SchematicLink feature class. Whereas the connectivity between HydroNetwork features is based on geometry, the links between schematic features are established using relationships. For example, a SchematicLink may be created to connect a SchematicNode representing a Watershed to the SchematicNode representing that Watershed's outlet, by following the relationship between a Watershed and its outlet HydroJunction. While the relationship may have been created using geometry, only the attributes defining the relationship are used to create the links between SchematicNodes. Thus, any type of feature (not just stream network features) may be represented in a schematic network.

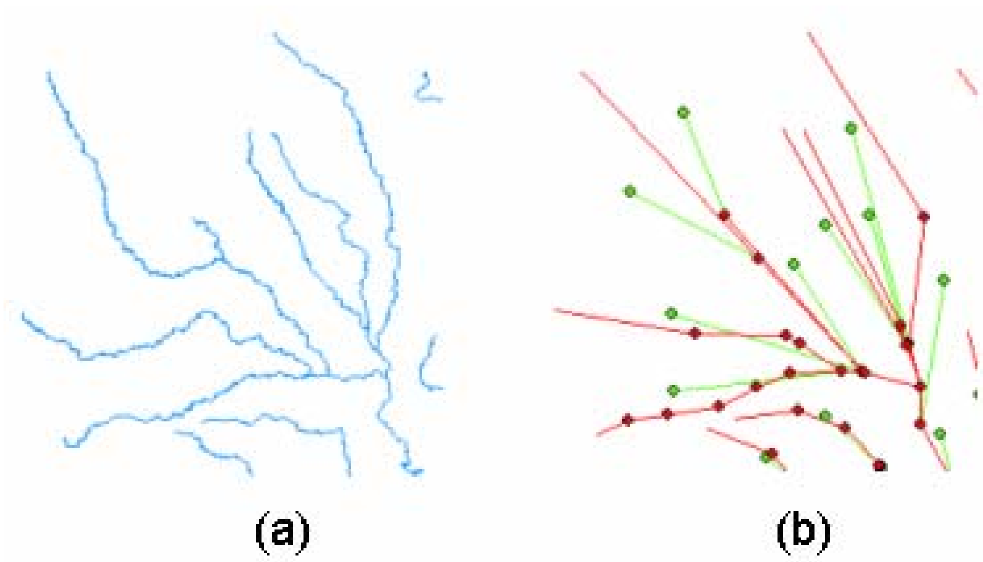


Figure 3.9 (a) HydroNetwork resembling cartographic representation of stream network, and (b) Schematic Network representation of hydrologic features, for watersheds upstream of Galveston Bay, TX

The Network package defines seven classes.

- **HydroEdge** – Linear network features, such as a river or stream
- **HydroJunction** – Point network features located at strategic locations, such as the outlet of watersheds
- **HydroNetwork_Junctions** – A generic junction created at the ends of all edges, except where other junctions already exist

- **SchematicNode** – A point representing a hydro feature in a schematic network, such as a monitoring point or watershed
- **SchematicLink** – A line connecting two SchematicNodes in a schematic network
- **HydroLineEvent** – A set of attributes describing a linear event on a line segment, defined by a range of measure values
- **HydroPointEvent** – A set of attributes describing a point event on a line segment, defined by a measure value

Relationships connect a HydroJunction to the Waterbody and Watershed for which it serves as the outlet, as well as the MonitoringPoints at that location.

3.3.4 Drainage

The Drainage feature dataset describes the drainage features of the landscape, typically derived from a digital elevation model. The Drainage feature dataset contains five feature classes.

- **Basin** – A drainage area defined for administrative purposes, such as a USGS HUC unit
- **Watershed** – A drainage area defined in a manner convenient for the current analysis
- **Catchment** – A drainage area defined by applying a set of rules to the landscape
- **DrainagePoint** – A point located at the center of the DEM cell serving as the outlet of a drainage area

- **DrainageLine** – A line created by traversing a DEM-derived drainage path

A relationship connects a Watershed to the HydroJunction that serves as its outlet.

3.3.5 Channel

The Channel feature dataset defines the channel system of a river. While raster data such as TINs may be used to describe a river channel, the Channel feature dataset only includes the vector representation of channels. The Channel feature dataset includes three classes.

- **CrossSection** – A 3-D linear feature representing a channel cross section
- **CrossSectionPoint** – A point on a cross section, typically defining some attribute at that location such as roughness or in some cases elevation
- **ProfileLine** – A 3-D linear feature representing the longitudinal profile of a river, such as the thalweg or bankline

A relationship connects a CrossSection to its CrossSectionPoints.

3.3.6 Time Series

Time series data are an integral part of hydrologic data. Arc Hydro uses two object classes to store time series data: TimeSeries and TSType. The TimeSeries class stores each individual timestamp and value pair as a record in a table. In addition to the timestamp and value, each record also stores a FeatureID and TSTypeID. The FeatureID identifies the feature that owns the time series record. The TSTypeID identifies the record in the TSType table that describes the type of the time series. The TSType table provides descriptive information about a set of time series records, such as the time interval or data origin. In this way, TSType acts as a lookup table for associating metadata with a set of time series records.

3.3.7 Arc Hydro Tools

The Arc Hydro toolset provides tools for creating or working with Arc Hydro data. The toolset is added to the ArcMap graphical user interface as a custom toolbar generated from a DLL. The Arc Hydro tools may be categorized into raster and vector tools. The raster tools process digital elevation models to produce vectorized watersheds, outlet points, and streams based on topography interpreted from the DEM. The vector tools populate the attributes of Arc Hydro, which entails determining topological relationships between features and calculating certain parameters for features. The vector tools also take advantage of the Arc Hydro data structure to perform attribute accumulation and attribute-based tracing routines (Whiteaker, 2001). A time series viewer displays time series data by rendering vector features based on the magnitude of their associated time series values, and stepping through each time step.

The Arc Hydro tools also maintain a unique identifier in Arc Hydro data called the HydroID. All features in Arc Hydro possess a HydroID, which uniquely identifies each feature in a geodatabase. The HydroID is used as a key field in all Arc Hydro relationships. For example, the FeatureID field in the TimeSeries class points to the HydroID of the feature that owns that time series.

The Arc Hydro tools were originally developed by the author. ESRI modified and added to the tools, and now publishes the official toolset on the ESRI web site <www.esri.com>.

3.4 NWIS

The USGS National Water Information System (NWIS) stores water resources data for 1.5 million sites in the United States and Puerto Rico. The NWIS database stores quantity and quality of surface and groundwater sites, in addition to descriptive information about each site. The data can be accessed online through NWISWeb at http://waterdata.usgs.gov/nwis. Real-time data are continuously added to the NWIS database, while some historical NWIS records date back more than a century (USGS, 2002b).

NWISWeb data are organized into five categories: Real-time, site information, surface water (quantity), ground water (quantity), and water quality. The data can be retrieved in tabular or graphical format. In the future, data may be retrievable in an XML format as well. There are more than 850,000 station years of surface water quality and quantity data on record. Data are recorded automatically or by manual methods and then relayed to USGS offices for storage, processing, and publication (USGS, 2002b).

Real-time Data

Real-time data may be retrieved for certain sites. The data are typically recorded at 15-60 minute intervals, and are current to within the past four hours. Real-time data can be retrieved for the past 31 days. Older records must be retrieved using the appropriate historical catalog.

A typical real-time time series record includes site number, timestamp, parameter names, parameter units, and parameter values. Examples of parameters include water temperature, specific conductance, dissolved oxygen, pH, precipitation, discharge, and gage height. There are more than 6000 parameters recorded (some categories have several entries, e.g. a parameter for each of eight sieve diameters for percent finer measurements). Data can be retrieved based on certain criteria, such as site number or name, state, hydrologic region, parameter(s) of interest, and number of days to retrieve. Statistical data can also be retrieved for certain measurements.

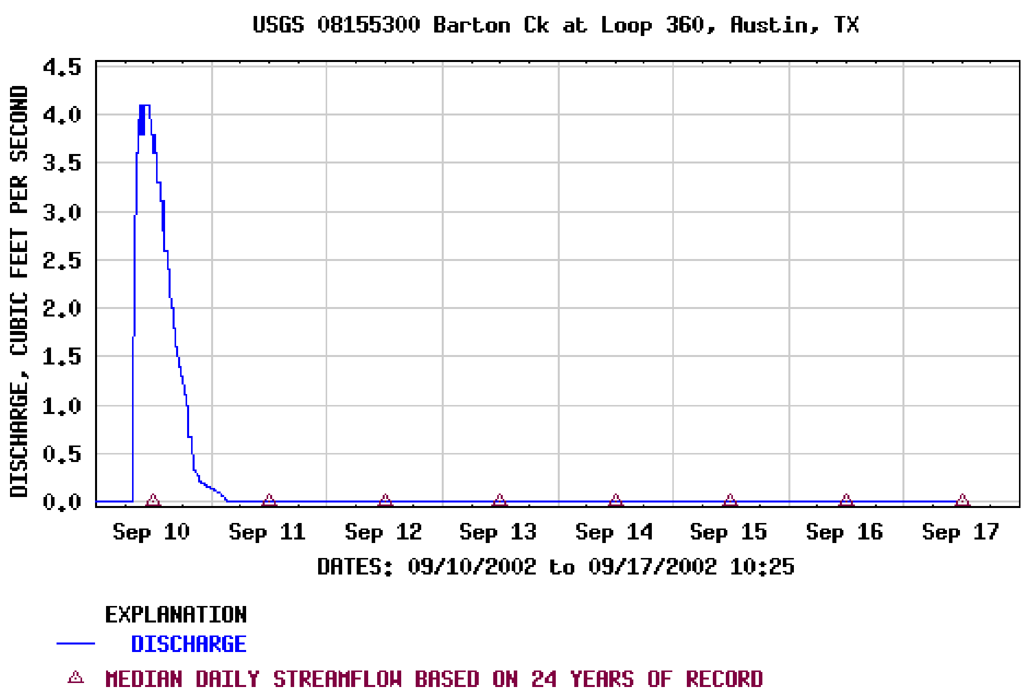


Figure 3.10 Example of Real-time Streamflow Data (USGS, 2002b)

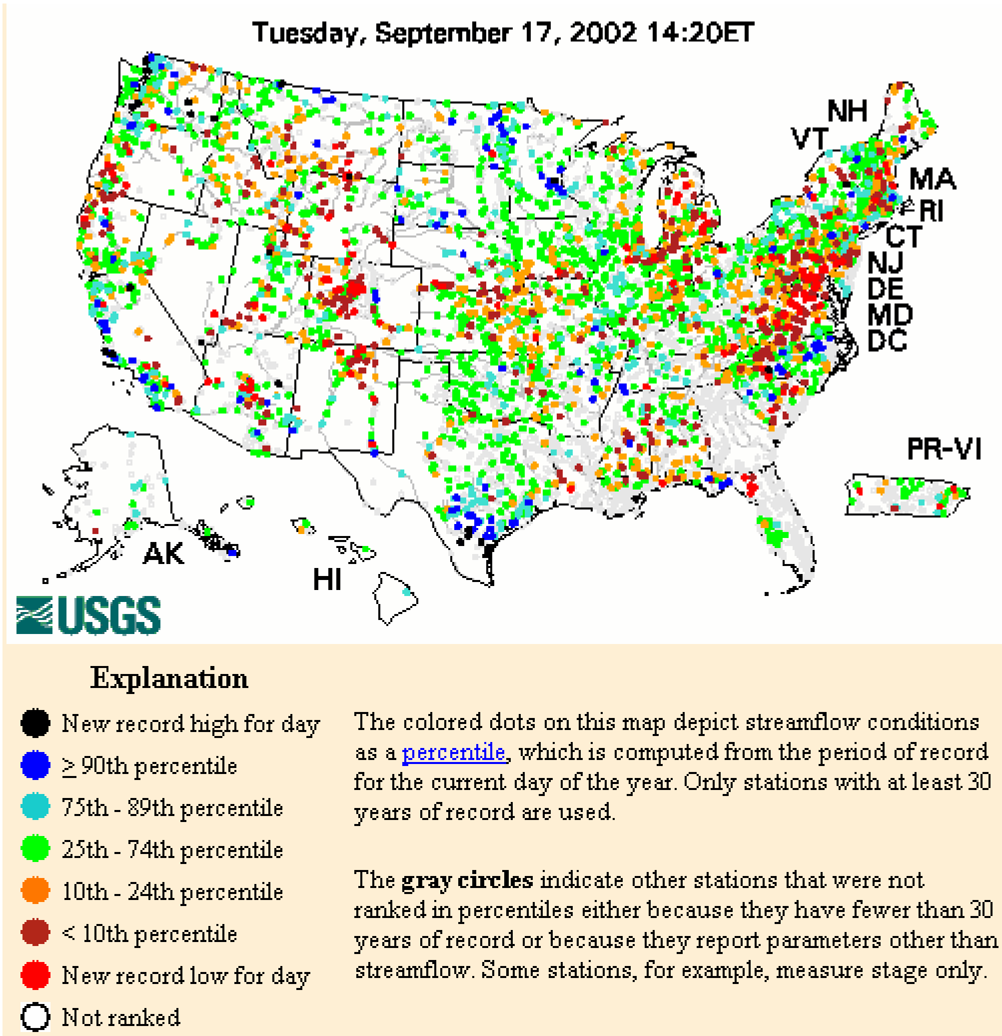


Figure 3.11 Current Streamflow Conditions in the United States (USGS, 2002b)

Site Information

Site information includes site identifier, geographic location, state, hydrologic unit code, parameters measured, time zone, and other data. The period of record for each parameter can also be obtained for a given station. Site information can be retrieved in tabular or XML format.

```
  <?xml version="1.0" ?>
- <usgs_nwis>
  - <site>
      <agency_cd>USGS</agency_cd>
      <site_no>08158000</site_no>
      <station_nm>Colorado Rv at Austin, TX</station_nm>
      <lat_va>301440</lat_va>
      <long_va>0974139</long_va>
      <lat_long_datum_cd>NAD27</lat_long_datum_cd>
      <state_cd>48</state_cd>
      <county_cd>453</county_cd>
      <alt_va>402.27</alt_va>
      <huc_cd>12090205</huc_cd>
    </site>
  </usgs_nwis>
```


Figure 3.12 XML Data for the Site on the Colorado River at Austin, TX (USGS, 2002b)

Surface Water

There are six categories of surface water data: real-time, recent, streamflow, statistics, peaks, and measurements.

- **Real-time** - Same as real-time data described above.
- **Recent** - At the end of each recording day, summary data are created and published for each site. These provisional data may be retrieved for the previous 18 months.
- **Streamflow** - After each water year, daily streamflow data are published.
- **Statistics** - Daily, monthly, and annual statistical data (such as mean flow) are computed from published daily data.
- **Peaks** - After each water year, peak flow data are published.

- **Measurements** - Field measurements are recorded for some gages. These measurements include streamflow, gage height, velocity, measurement type, as well as some qualitative information about the nature of flow.

Number	Date	Made By	Mean Vel	Stream flow	MS Type	Control
1570	2002-07-23 09:00	SMJ/RAS	2.32	6740	CABLEWAY	LGT DEBRIS
1569	2002-07-19 09:14	RAS/KCW	2.57	10900	BRG CRANE	CLEAR

Table 3.2 Example of Measurement Data for the Colorado River at Austin (USGS, 2002b)

Groundwater

The groundwater database contains groundwater level and water quality data for wells, springs, tunnels, excavations, and other features. There are three categories of groundwater data: real-time, site information, and levels.

- **Real-time** - Same as real-time data described above.
- **Site Information** - Same as site information described above.
- **Levels** - Contains water-surface elevation or depth to water data in wells.

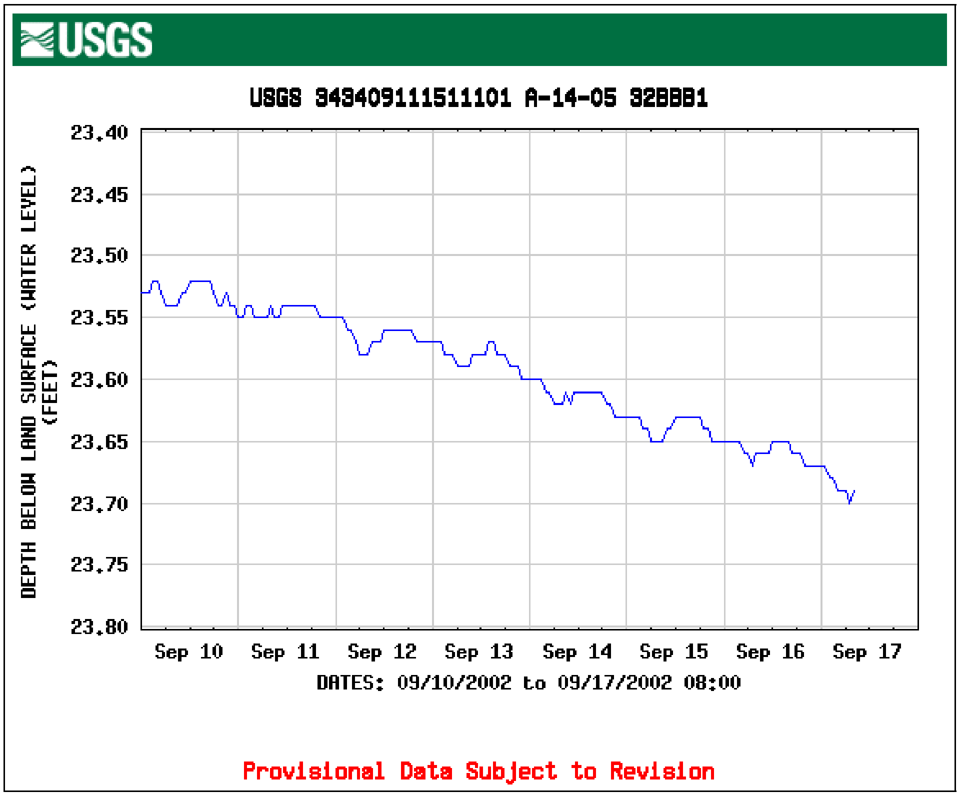


Figure 3.13 Real-time Groundwater Data in Arizona (USGS, 2002b)

Water Quality

The water quality database stores chemical, physical, and biological properties of water, sediment and tissue samples. Real-time and archived data can be retrieved. The listing of available data includes dates recorded, type of sample, and number of samples. While this database contains over 3.5 million historical water quality analyses, the site recommends looking at EPA STORET for more water quality information.

SAMPLE DATETIME	MEDIUM CODE	TEMPER-ATURE WATER (DEG C) (00010)	PH WATER WHOLE FIELD (STAND-ARD UNITS) (00400)	NITRO-GEN, ORGANIC DIS-SOLVED (MG/L AS N) (00607)
1973-10-16 12:35	9	15.5	7.2	
1975-01-13 17:02	9	6.5	5.0	.34

Table 3.3 Example of Water Quality Data for Jacks Swamp near Pleasant Hill, North Carolina (USGS, 2002b)

The retrieval of water quality data, and all other data available on NWISWeb, is based on parameters specified in a URL. A URL (uniform resource locator) identifies an object on the Internet, such as an ftp file, a newsgroup, or a web page. NWISWeb uses URLs to define web pages for displaying data. The URL contains information about the site, data type, and period of record for data to retrieve from the NWIS database, as well as formatting instructions. NWISWeb interprets the URL to query the NWIS database and build a web page containing the requested information.

Typically, a user accesses various web pages on NWISWeb in order to identify the monitoring station of interest, the type of data to retrieve, and the period of record to retrieve. The web site then builds the URL for the user and queries the NWIS database for the data using the URL. But because all necessary parameters for data retrieval are specified in the URL, data may be retrieved without navigating through the NWIS Web site. If a user knew the format that

NWISWeb expects from the URL, the user could type in the URL manually, providing the web site everything it needs to retrieve the data. A more enticing benefit is that programs familiar with the NWIS URL format can retrieve data automatically without having to use a web browser. The request for data retrieval can be sent from a simulation model or a GIS. An example NWIS URL is shown below. This URL is used to retrieve daily streamflow data for the Colorado River at Austin, TX, for 1999, in tab-separated format.

http://waterdata.usgs.gov/nwis/discharge?site_no=08158000&agency_cd=USGS
&begin_date=1999-01-01&end_date=1999-12-
31&set_logscale_y=1&format=rdb&date_format=YYYY-MM-
DD&rdb_compression=&submitted_form=brief_list

Figure 3.14 Sample NWIS URL

NWISWeb would return the following web page from the given URL.

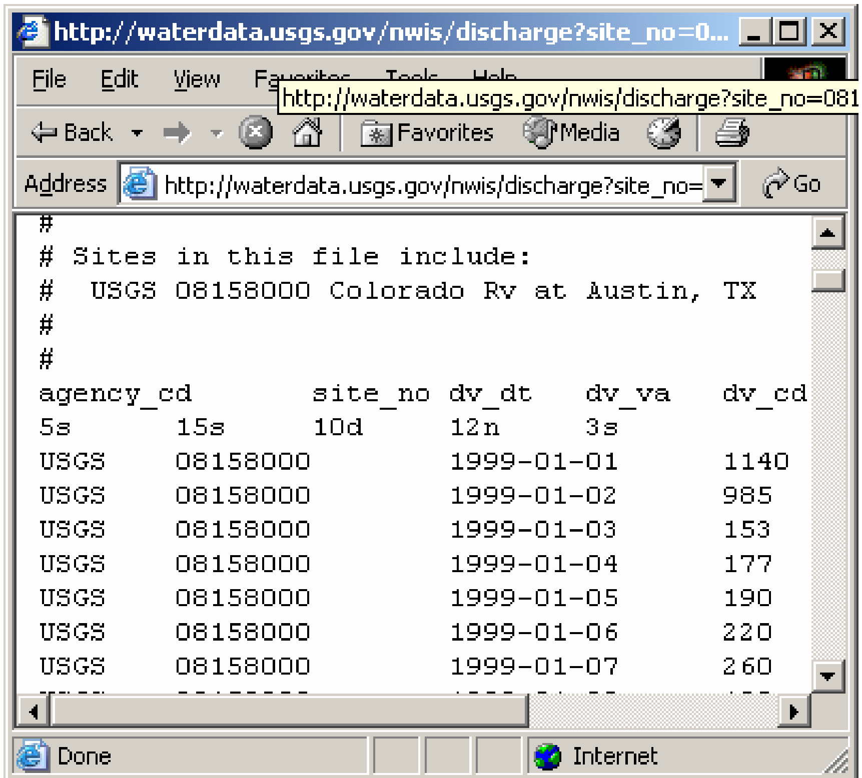


Figure 3.15 NWIS Daily Streamflow Data

While not providing simulation models, NWISWeb does provide tools to access and display a huge database of national water quality and quantity information. Because data can be retrieved from NWIS simply by creating the appropriate URL (which is essentially a string of text), NWIS data can be easily incorporated into any application that can access the Internet.

3.5 CONCLUSIONS FROM TECHNOLOGY REVIEW

The programming techniques presented in this chapter provide useful guidelines, which are utilized in the technical products developed from this research. ArcGIS serves as a flexible platform for handling geospatial and temporal data. This research will utilize new ArcGIS 9 geoprocessing framework to assemble hydrologic simulation models into a workflow sequence along with geoprocessing tasks to form a complete hydrologic modeling system. The ArcGIS Hydro data model provides the data structure that makes the integration of simulation models and GIS possible. Data from websites such as NWIS can be used as simulation model inputs, or to verify simulation model results.

Chapter 4 Raster-Network Regionalization Technique

This chapter discusses the concept of Raster-Network Regionalization. Regionalization refers to the integration of subregional datasets into regional datasets for analysis purposes. First, problems with raster-based analysis of large datasets are outlined, followed by a discussion of how Arc Hydro and network-based accumulation improves upon raster techniques. With this background, the Raster-Network Regionalization Technique is then described. Finally, a case study utilizing Raster-Network Regionalization to prepare GIS data for use in the WRAP simulation model is presented.

4.1 PROBLEMS WITH RASTER-BASED ANALYSIS

Raster-based analysis of GIS data provides useful parameters for hydrologic modeling, water quality assessments, and other water resources applications (Mason, 2000; Osborne, 2000). Distributed parameters can be summarized and accumulated using a variety of raster tools available through the GIS. However, problems arise as the size of the raster datasets increase, as when working in with very large areas or with very high-resolution data. These problems include:

1. The time required to process rasters becomes unreasonable. Figurski (2001) reported processing times on the order of 10 days for calculating watershed parameters for the Trinity Basin at 30-meter raster resolution (over 100 million cells). This problem is compounded when errors occur during the processing (imagine a crash on the 9th day!) or when new

information must be incorporated into the analysis, requiring the process to start over.

2. Data storage becomes cumbersome. The calculation of multiple grids covering a vast number of cells leads to a large amount of data that must be stored on disk. In extreme cases, disk space may run out, or data management and manipulation may become impossible. Additionally, the transfer of such information to other parties (such as clients) also becomes cumbersome.
3. Required grid size exceeds limits. In ArcGIS 8, grids are limited to a maximum size of 2 gigabytes. If a raster analysis required a grid with a resolution sufficient to exceed the 2 gigabyte limit, then that raster analysis cannot be performed.

Despite these drawbacks, the distributed analysis capabilities of raster processing are too valuable to discard. Figurski (2001) developed a technique of cascading parameters, so that rasters could be split into parts. However, the technique still relied on rasters for accumulation routines, and was prone to error due to the large number of hand manipulations of data involved. A more sophisticated network-based accumulation routine is described in the next section.

4.2 Arc Hydro and Network-based Accumulation

Whiteaker (2001) developed the Accumulation and Consolidation tools for Arc Hydro, which operate on Arc Hydro data. These tools use relationships between features to accumulate information from one feature to another. For example, consider the Watershed and HydroJunction feature classes from Arc Hydro. The JunctionID attribute on Watershed identifies the HydroID of the HydroJunction that serves as that Watershed's outlet. The Consolidation tool can be used to calculate the total area of Watersheds draining to a particular HydroJunction. First, the tool identifies Watersheds with a JunctionID matching that HydroJunction's HydroID. Then, the AreaSqKm attribute from each Watershed is read and added to a total DrainArea value, which is written to the DrainArea attribute in HydroJunction.

Figure 4.1 Area from Upstream Watersheds Consolidated in Outlet Junction

The Accumulation tool is designed to pass attributes downstream among HydroJunctions in a stream network. Downstream navigation is performed by reading the NextDownID attribute of a given HydroJunction, which points to the HydroID of the next downstream HydroJunction in the stream network. Each HydroJunction's value (e.g. DrainArea) is added to the next downstream HydroJunction's value, so that the final DrainArea value for a given HydroJunction contains the DrainArea from all upstream HydroJunctions. The Accumulation tool is slightly more complicated than the Consolidation tool, because the Accumulation tool must navigate through a series of NextDownID-HydroID associations until all upstream HydroJunctions have been accounted for, for a given HydroJunction; whereas the Consolidation tool only navigates a single ID association for each feature. Also, the Accumulation tool is designed to work within the same feature class, while the Consolidation tool is designed to work with two different feature classes.

An important caveat for the accumulation tool is that the tool is designed to work with dendritic stream network. If branching occurs, each side of the branch may not be accounted for correctly, as only one value for NextDownID can be stored for a given HydroJunction, even if that junction is upstream of two other junctions on either side of a branch.

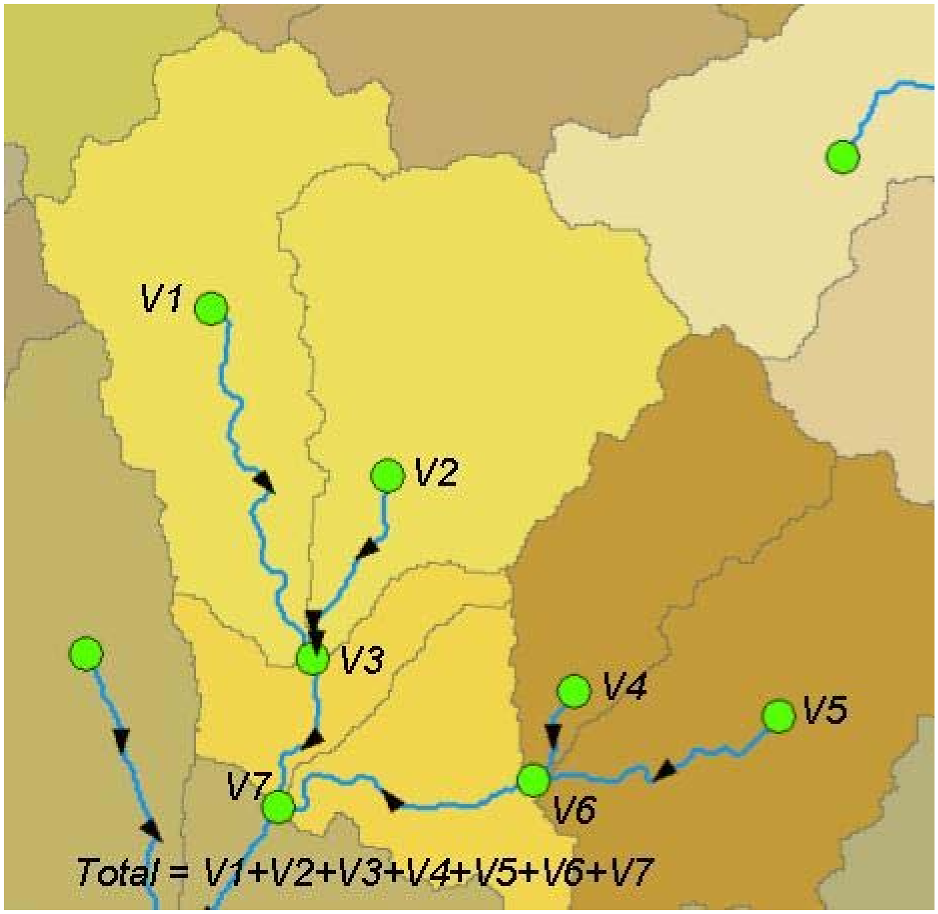


Figure 4.2 Accumulated value for the outlet junction is the sum of all upstream values plus the value at the outlet, or V1+V2+V3+V4+V5+V6+V7

4.3 THE RASTER-NETWORK REGIONALIZATION TECHNIQUE

The Accumulation and Consolidation tools allow for rapid summation of attributes in the vector domain. Not only is vector accumulation typically faster than computing large raster datasets, but the size on disk of vector data required to describe the entire basin is typically much less than the amount of space required to store flow-accumulated rasters covering the study area.

For example, consider the Guadalupe River Basin in Texas, which covers 15,462 square kilometers, or roughly 17 million raster cells at a 30 meter by 30 meter cell size. As a test case, this basin was divided into 2035 Watersheds, with an outlet HydroJunction on the stream network for each Watershed. The accumulation and consolidation tools were used to attribute HydroJunctions with total upstream drainage area. On a 2 GHz computer with 512 MB RAM, the process takes 22 minutes. To computer the same values using raster methods, a flow accumulation grid must be created. The creation of this grid from a flow direction raster takes 1 hour 43 minutes, or nearly five times as long as using vector means. Additionally, the flow accumulation raster consumes 281 MB of disk space. Each additional parameter to be accumulated (such as average precipitation or curve number) would require a new weighted flow accumulation grid, and would take up approximately the same amount of disk space. While the geodatabase containing the vector data for the accumulation and consolidation routines requires 28 MB of disk space, watershed and stream network data are typically created for water resources data anyway. Therefore, the additional space required to use the vector tools is simply the space required to add the fields to store the attributes on the Watersheds, or roughly 0.04 MB in the case of the Guadalupe data.

Yet despite the efficiency in using network-based analysis techniques, some raster analyses must still be performed. So, how can the Accumulation and Consolidation routines be applied to geospatial processing for water resources applications?

The key is in the Watersheds. Often, hydrologic parameters are defined and calculated for watersheds. For example, in the traditional method of calculating parameters for the WRAP simulation model in a GIS, control points (points of interest) are overlain on a flow accumulated raster. A grid cell underneath a given control point is read to determine the value for that control point. The cell represents a weighted flow accumulation for a certain value, such as average curve number, which includes the influence of all grid cells which flow (hydrologically) to that grid cell. Together, all of those cells form the watershed for that point of interest. Therefore, by reading the value of the cell underneath a given control point, the average value for the watershed draining to that control point is determined.

The *Raster-Network Regionalization Technique* uses a different approach. A summarization of raster values over watersheds can be easily determined by using the watersheds as distinct *zones* which define the area of analysis for the zonal statistics tool in ArcGIS. This tool calculates statistics such as mean, sum, max, and min for each zone by reading the values of cells within each zone and performing the necessary statistical operations. Thus, with this approach, accumulated grids whose cell values are influenced by all upstream cells are no longer needed. The only cells that a watershed is interested in are the cells that lie directly over that watershed.

Once attribute values have been determined for watersheds, these values can be transferred to outlet junctions, and then consolidated throughout the stream network in the vector domain. The watersheds become the basic processing unit with basin-wide coverage, while raster coverage can be reduced to each individual watershed's extent. Thus, watersheds effectively replace grid cells as the 'units' of analysis.

This allows a basin or region to be divided into hydrologically distinct subregions, in which the necessary raster analyses takes place. The smaller size of the subregions permits faster raster processing, while results from raster analyses are stored on vector Watersheds to be accumulated at the basin level. The Consolidation and Accumulation techniques described above are then used to accumulate Watershed parameters across the entire basin.

The HydroID, a unique feature identifier in an Arc Hydro geodatabase, plays an important role in regionalization. The HydroID of each feature in a subregion is prefixed by a number identifying to which subregion that feature belongs. Thus, when regional data are merged for the entire basin, one can still easily keep track of which subregion a given feature lies within.

The Raster-Network Regionalization Technique involves the following steps:

1. Clip raster and vector data to hydrologically-distinct subregions. Subregions may be defined by using established watershed boundaries such as USGS HUC boundaries.
2. Perform raster processing on each subregion, to obtain necessary values summarized onto vector watersheds.

3. Merge the subregional vector data to a regional vector dataset, and update connectivity for the outlet junction of each subregion to point to the nearest junction in the next downstream subregion.
4. Use the Accumulation and Consolidation routines to compute final values for each point of interest on the stream network.

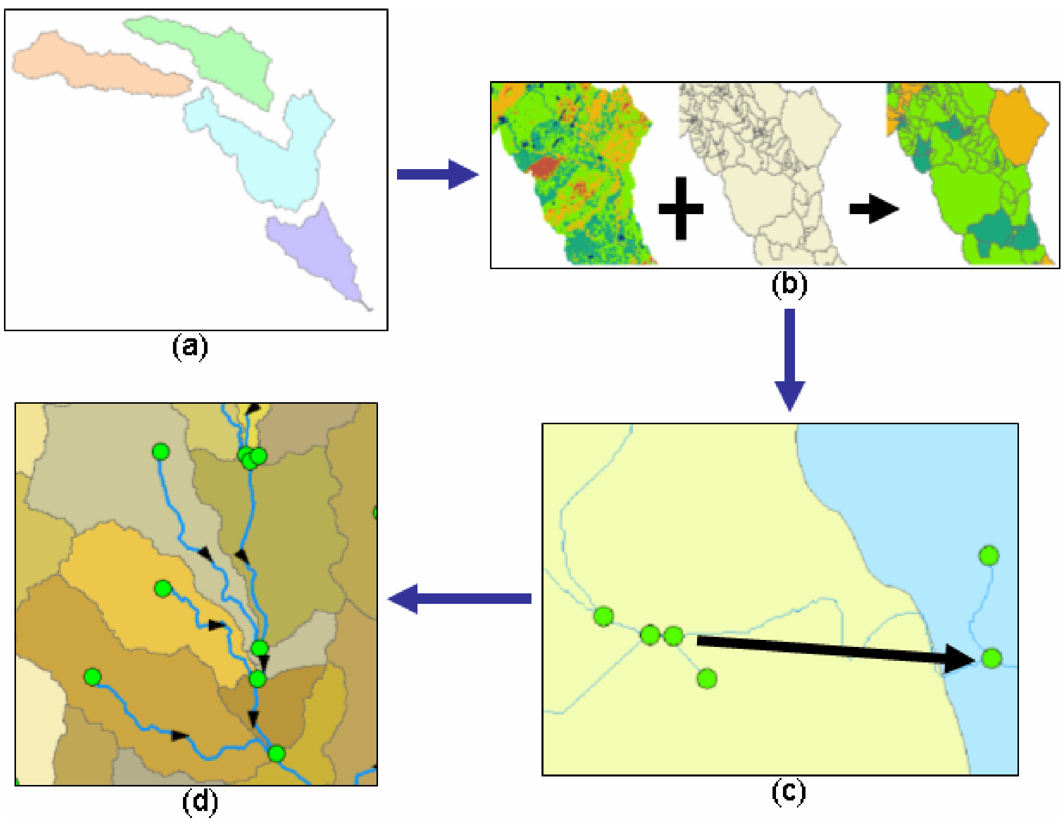


Figure 4.3 Steps in the Raster Network Regionalization Technique: (a) Defining subregions, (b) Performing subregional raster analyses, (c) Merging vector subregions, (d) Accumulating vector attributes using the stream network

The benefits of the Raster-Network Regionalization approach include:

1. Processing time is reduced, as much of the processing occurs in the vector domain rather than in the raster domain.
2. Data storage requirements are lessened, as accumulation grids no longer need to be created.
3. The remaining grids can be split into hydrologically-distinct regions defined by one or more watersheds. This allows for faster processing on the raster side, more modular data storage, and less of a raster reprocessing effort if data in a given watershed changes (since only those cells within that watershed matter to that watershed.)

With Raster-Network Regionalization, the weight of processing is shifted from the raster side to the vector side, resulting in several benefits. Even the largest basins can now be processed with high-resolution raster data. The Raster-Network Regionalization Technique was developed by Tim Whiteaker and Hema Gopalan. More information about Raster-Network Regionalization can be found in Gopalan (2003).

4.4 Case Study: WRAP Hydro

WRAP Hydro is a preprocessing data model created to provide the Water Rights Analysis Package (WRAP) with geospatial inputs regarding watershed parameters and water rights connectivity. In addition to the data model, WRAP Hydro also consists of a toolset called the WRAP Hydro tools, which calculates those geospatial parameters for WRAP. WRAP Hydro utilizes the Raster-Network Regionalization Technique to handle large river basins. The WRAP simulation model, in the context of water availability modeling, is described in Chapter 2.

4.4.1 WRAP Hydro Data Model

WRAP Hydro utilizes a GIS to store geospatial inputs required to calculate parameters for WRAP. WRAP Hydro is not an Interface Data Model, as the direction of communication is one-way from the geodatabase to the WRAP simulation model. Rather, WRAP Hydro is an Arc Hydro-based Preprocessing Data Model. Arc Hydro serves as the basis for several WRAP Hydro feature classes; therefore, data prepared for use in WRAP may still be utilized in other simulations by navigating back through Arc Hydro, and then through the appropriate Interface Data Model.

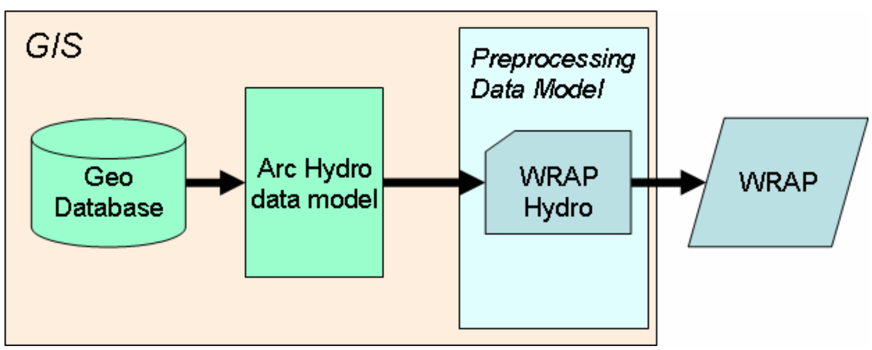


Figure 4.4 WRAP Hydro Permits One-Way Communication with WRAP

WRAP Hydro data are divided into separate geodatabases and folder locations for each region in the analysis. Data for a given region are stored in a single geodatabase called WRAPHydro*i*, where the *i* at the end of the name corresponds to the region number for the geodatabase. The database contains four feature datasets: ArcHydroRegion, BaseData, PreProcess, and WRAPHydro.

The *ArcHydroRegion* dataset stores Arc Hydro data for the region of interest. For example, NHD river reaches may be downloaded and stored in the HydroEdge feature class for the given region. *BaseData* stores WRAP-specific base data, including existing and proposed water rights locations. The *PreProcess* dataset is used to manipulate base and regional data so that WRAP parameters may be calculated. Preprocessing tasks include conditioning of raster data and strict definition of the basin boundary. Once preprocessing is completed, WRAP parameters are calculated in the WRAPHydro feature dataset. From there, the results may be exported for use in the WRAP simulation model, or combined with data from other regions.

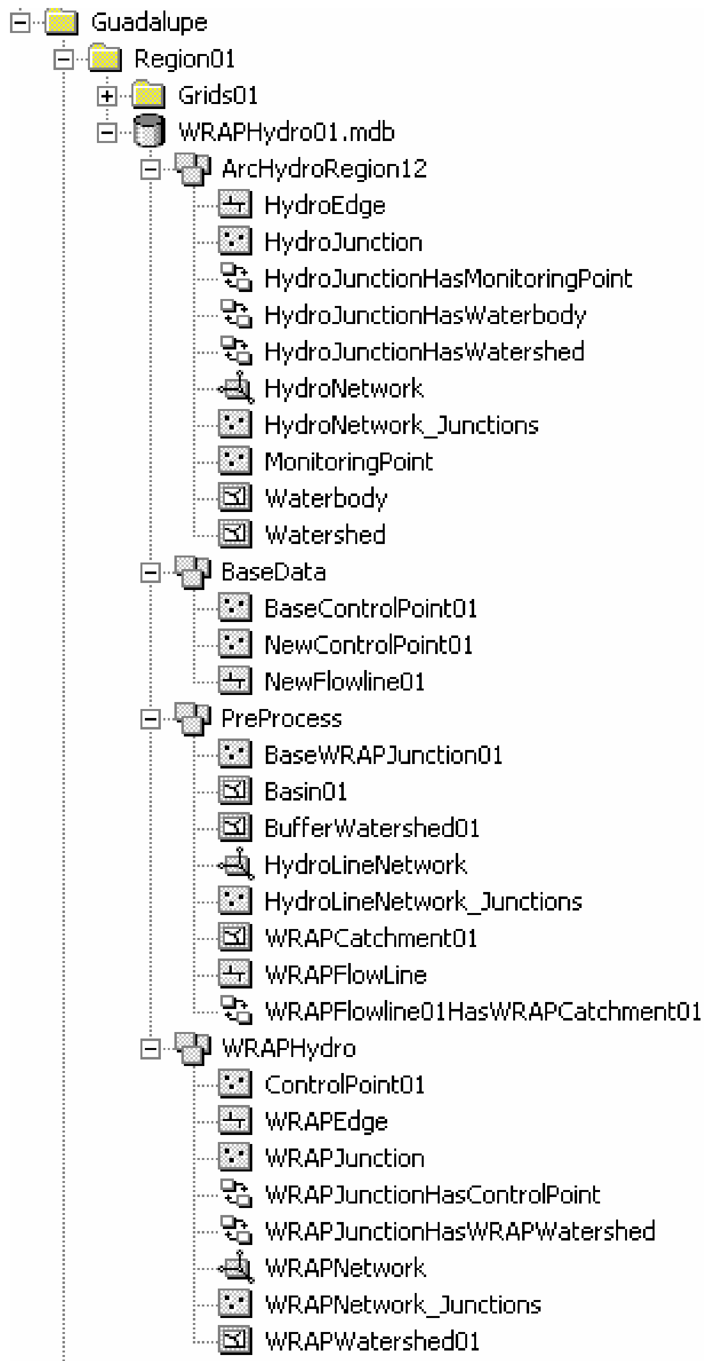


Figure 4.5 WRAP Hydro Geodatabase Structure

In addition to the vector data, a Grids folder exists in the same directory as the geodatabase to store raster data used in WRAP Hydro. The Grids folder contains three folders: BaseGrids, PreProcessGrids and WRAPHydroGrids. These folders store the grids used or developed in each process of WRAP Hydro data development.

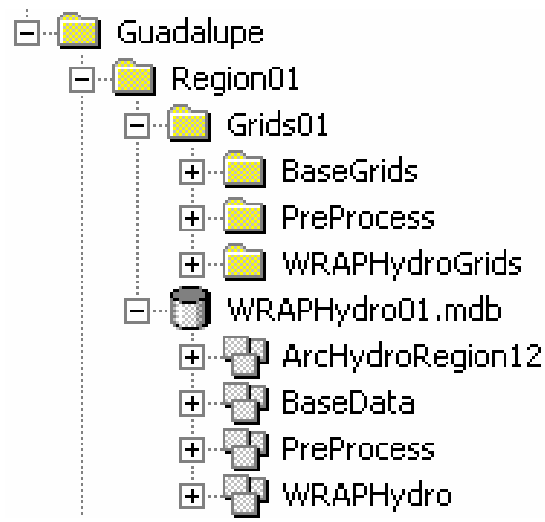


Figure 4.6 WRAP Hydro Geodatabase and Grid Folders

Together, the Grids folder and the WRAP Hydro geodatabase describe data for a particular region. Each region's data in a WRAP Hydro analysis is stored in its own folder. Once regional data have been processed, the vector data are merged into a single geodatabase located at the root level of the WRAP Hydro folder structure. Because all preprocessing has already been performed in the regional datasets, only a single WRAPHydro feature dataset is needed in the merged geodatabase.

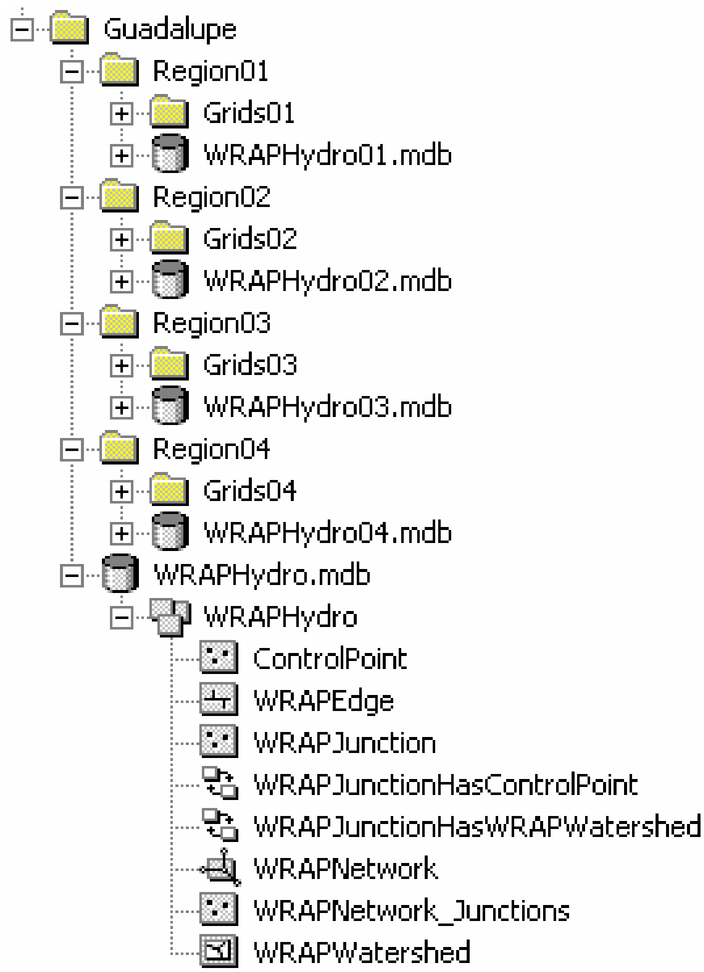


Figure 4.7 WRAP Hydro Folder Structure for Four Regions, Including WRAPHydro Geodatabase with Merged Results

The actual processing of WRAP Hydro data is facilitated by the Arc Hydro and WRAP Hydro tools. The resulting data contain all of the geospatial inputs required by the WRAP simulation model.

4.4.2 WRAP Hydro Tools

The WRAP Hydro tools (developed by the author) consist of a set of public domain utilities developed on top of the Arc Hydro data model. These tools operate in the ArcGIS ArcMap environment, with some of the functions requiring the Spatial Analyst extension. The tools provide functionality in addition to the Arc Hydro tools in order to calculate drainage area, average curve number, average precipitation, and the next downstream identifier for controls points in WRAP.

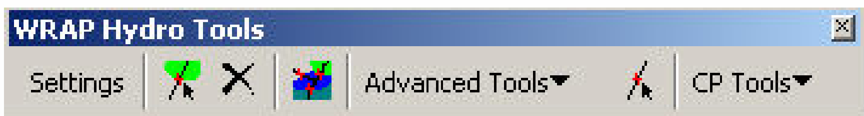


Figure 4.8 WRAP Hydro Tools ArcMap Toolbar

This section describes some of the core functionality of the WRAP Hydro tools and how the tools are used to calculate WRAP parameters. Full help documentation for the tools, as well as the tools themselves, may be downloaded from http://www.ce.utexas.edu/prof/maidment/grad/whiteaker/hydrotools.html.

Watershed Delineation

The WRAP Hydro tools include utilities for delineating watersheds, which allow the user to choose a feature class of point, line, or polygon geometry to serve as the outlet zones for the watersheds. Traditionally, watersheds are delineated in a GIS from a set of input points and a flow direction grid. The GIS uses the flow direction grid to determine which cells flow to a given input point before any other input point. That set of cells is merged to produce a watershed feature for the given input point. Thus, each point defines an outlet in the resulting watershed feature class.

The weakness of this approach occurs when the point is not placed in the proper location over the flow direction grid. If the point is not over a cell within a natural channel in the digital elevation model (or where a stream has been burned in), then the watershed that is delineated for that point will not define the intended drainage boundary.

A more secure approach is to attribute the edges that drain to a particular junction with the ID of that junction, and then to delineate watersheds to the edges. As the watershed outlets, there is a much greater chance that the edge geometries will intersect a cell in the channel of the DEM. The watersheds may be merged using the attribute from the edge feature class which stores the ID from the outlet junction feature. Thus, one watershed will be created for each outlet junction, with each watershed already possessing an attribute identifying its outlet junction.

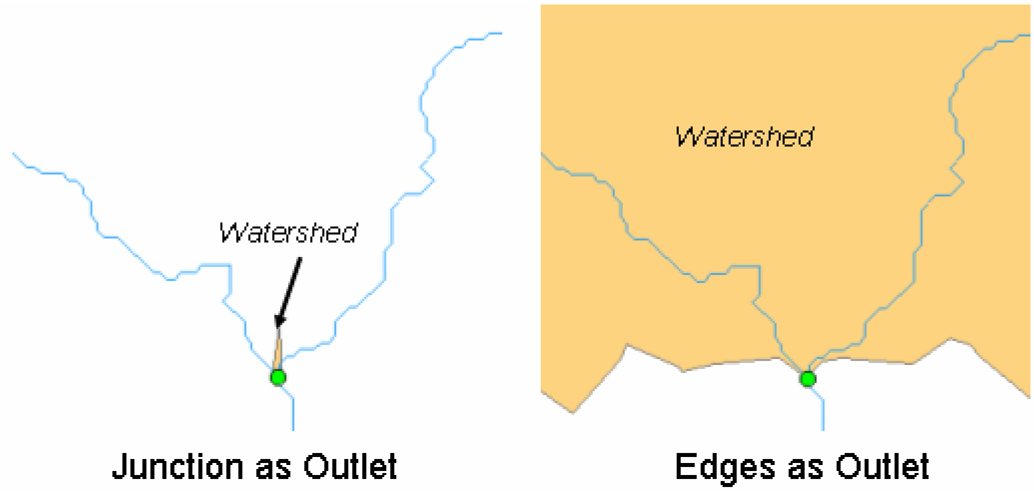


Figure 4.9 Using different features as outlet zones for watershed delineation yields different results (only a portion of the watershed formed from using edges as outlets is shown)

The WRAP Hydro tools give the user the capability of choosing which feature class to use as the watershed outlets for delineation purposes. This feature class is called the source layer. The source layer may contain point, line, or polygon features. This approach becomes particularly useful in flat areas, where the flow direction is more ambiguous.

Parameter Calculation

Once watersheds have been delineated, the WRAP parameters of average curve number, average precipitation, and drainage area may be calculated using the WRAP Hydro tools. The drainage area is simply read from the shape area of the watershed feature, which is calculated automatically by ArcGIS, and then converted to desired units of analysis. Average curve number for a given watershed is found by laying a curve number grid over a watershed and then using zonal statistics to computer the average curve number value for all cells over that watershed. The same process is used to calculate average precipitation for watersheds.

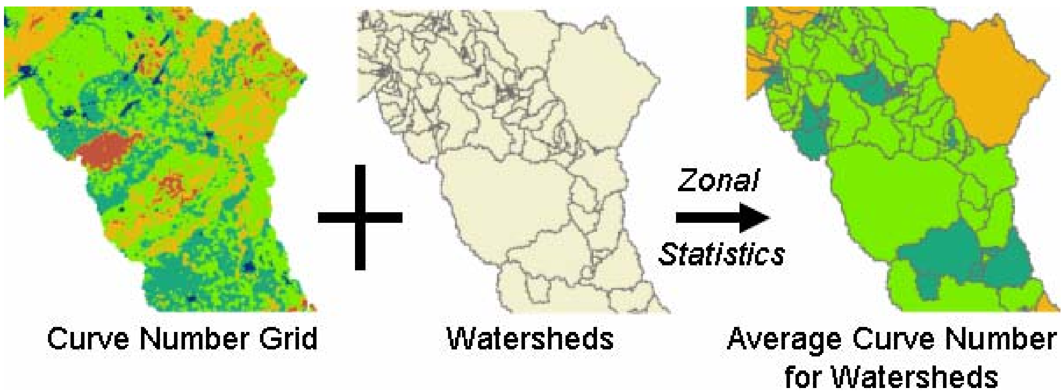


Figure 4.10 Zonal Statistics Provide Average Curve Number for Each Watershed

Once parameter values have been calculated for each watershed, they can be accumulated onto control points using the consolidation and accumulation techniques described above. The WRAP Hydro tools use those routines without modification to accumulate drainage area values. For average curve number, each watershed's curve number value is multiplied by that watershed's area. This weighted curve number is then accumulated for the control points. The final accumulated value is divided by the total combined area of upstream watersheds to provide an average curve number for the entire drainage area of a given control point (Eq. 4.1).

$$CN_j = \frac{\sum_{i=1}^{n} A_i CN_i}{\sum_{i=1}^{n} A_i} \qquad \text{(Eq. 4.1)}$$

Where:

CN_j = Average curve number for control point *j*

A_i = Drainage area for *i*th watershed draining to control point *j*

CN_i = Curve number for *i*th watershed draining to control point *j*

n = Number of watersheds draining to control point *j*

The average precipitation for each control point is found in the same manner. Once these attributes have been calculated, the results may be exported for use in the WRAP simulation model.

With its well-organized data model and custom toolset, WRAP Hydro provides a useful mechanism for managing Water Availability Modeling (WAM) data, and calculating parameters required to run WRAP simulations. The Texas Commission on Environmental Quality has adopted WRAP Hydro for its WAM projects, and is currently cooperating with the University of Texas in processing the Rio Grande River Basin. The Raster-Network Regionalization Technique is vital to that effort, as the Rio Grande comprises an area of 470,000 km^2, or well over 500 million grid cells at 30-meter resolution.

This case study has illustrated an example of an Arc Hydro-based Preprocessing Data Model and application of the Raster-Network Regionalization Technique. Both of these methods of geospatial integration facilitate the use of a

GIS to prepare input parameters for simulation models. Full details of WRAP Hydro, including the role of each feature class in the data model and further elaboration on the use of the WRAP Hydro tools, may be found in Gopalan (2003).

Chapter 5 Model Integration Through Exchange of Time Series at Information Exchange Points

A GIS can provide a useful medium for the exchange of time series information between simulation models. While the internal physical descriptions of hydrologic features vary between models, certain basic types of features are common throughout models, such as rivers and junctions on rivers. Arc Hydro captures the core essence of those features in its data model, and thus may serve as a 'common ground' between hydrologic simulation models.

This chapter describes a method of exchanging time series information between simulation models at *information exchange points* on the stream network. A case study implementing the methodology for a floodplain mapping application for Rosillo Creek in the San Antonio River basin is presented.

5.1 METHODOLOGY

Simulation models have been tested and proven through years (sometimes decades) of use. Today's water resources engineering problems demand solutions that utilize output from more than one type of simulation model. For example, a real-time flood forecasting model not only requires a hydrologic component to convert rainfall to runoff, but also a hydraulic component to route the flow through the stream network and predict the timing and severity of the flood wave. Direct integration of different models is very difficult, if not impossible. Such a task may require manipulation of the source code in the models themselves, as well as a reconciliation between time series structures and feature representations between the models.

Geospatial integration offers a more feasible approach. In geospatial integration, a hydrologic information system (HIS) is constructed, which contains a central GIS database and the simulation models required for the analysis. Each hydrologic model's execution occurs independently of other components in the system, while the outputs of a model simulation are imported into the GIS in order to be used as input to another simulation model or for further geospatial analysis and interpretation of results. Because the focus of this methodology is on water resources applications, the exchange of information between simulation models only occurs at information exchange points within the GIS, where an information exchange point is defined as any point of interest which holds significance regarding the flow of water over the land surface. These points are typically located at HydroJunctions, which may be linked to watersheds, cross sections, and other features through relationships. The types of information that may be exchanged at information exchange points include time series and attribute information, such as streamflow and water quality loads.

The information exchange occurs between model simulations, rather than during a given simulation. While this approach may not be as powerful as fully coupled, simultaneous execution of simulation models, it is much easier to implement, and still provides useful and flexible solutions for situations in which the simulation models may be executed sequentially, such as with rainfall-runoff-routing applications. Additionally, because the models are not hardwired directly with each other, different models may substituted to simulate a given hydrologic or hydraulic process, so long as an interface for that model has been created to communicate with the GIS. The following case study illustrates the above methodology.

5.2 CASE STUDY: CONVERTING NEXRAD DATA TO FLOOD INUNDATION POLYGONS

5.2.1 Overview

The purpose of the application described in this case study is to convert from NEXRAD rainfall data to flood inundation polygons for Rosillo Creek, a tributary of the San Antonio River in Texas. The Rosillo Creek basin covers an area of 73 square kilometers, and exhibits a short response time to rainfall events, on the order hours. The automated procedure for this case study involves converting the rainfall time series to runoff hydrographs for each watershed in the Rosillo Creek basin using HEC-HMS, taking those hydrographs at the outlet for each watershed to serve as inputs to a HEC-RAS hydraulic model, and performing GIS processing to convert the resulting cross section water surface elevations to a polygon of inundated area. The application is implemented using ModelBuilder in ArcGIS 9. The application will support the San Antonio River Authority's floodplain modeling efforts. Oscar Robayo and Tim Whiteaker at the University of Texas at Austin were the primary developers of the application, with assistance from Zichuan Ye and Nawajish Noman at ESRI.

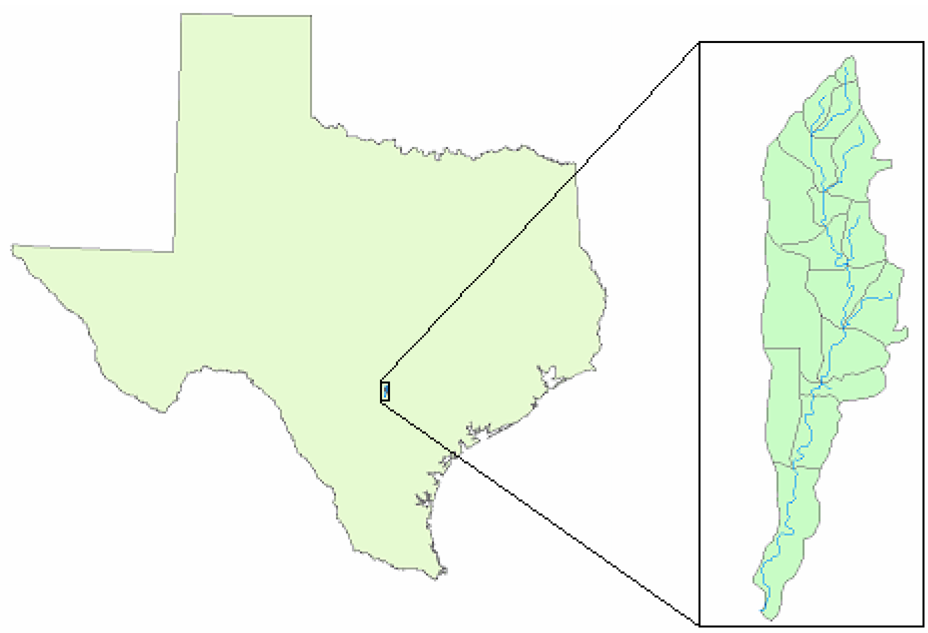


Figure 5.1 ROSILLO CREEK IN TEXAS

5.2.2 Data Model

This case study involves two simulation models: HEC-RAS and HEC-HMS. Thus, an Interface Data Model was created for each simulation model. The HMS Interface Data Model contains Watersheds and SchemaNodes with an HMSCode attribute. This attribute links features in the geodatabase to their representation in an HMS Basin file.

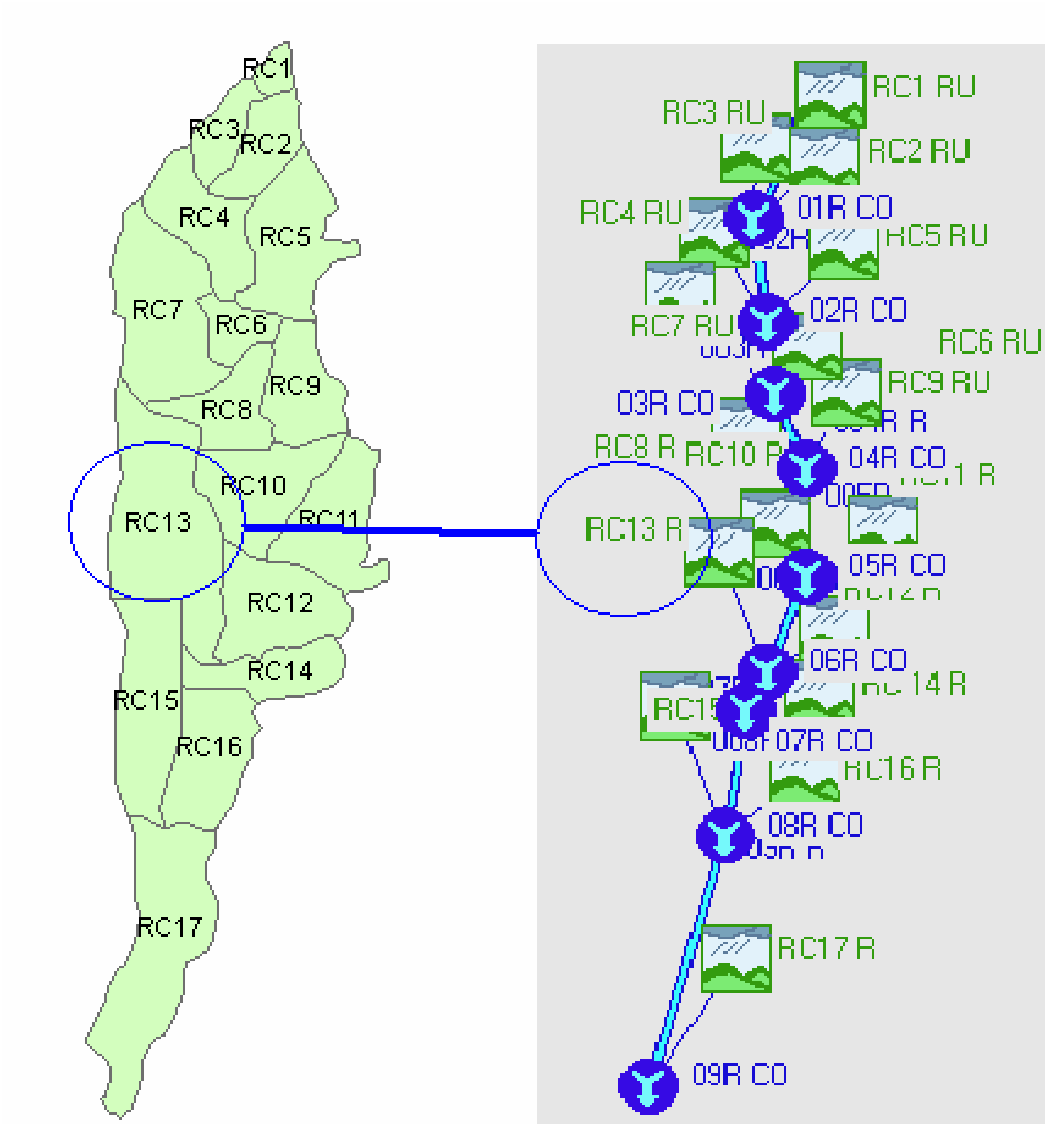


Figure 5.2 FEATURES IN THE GEODATABASE AND HMS ARE LINKED THROUGH HMSCODE

The RAS Interface Data Model extends the Arc Hydro CrossSection feature class to include Stream_ID, Reach_ID, and Station attributes. These attributes locate a given cross section in a RAS model. In addition to those feature classes required by the RAS simulation model, the RAS Interface Data Model also includes a Boundary feature class. This feature class is taken from an ArcGIS extension called GeoRAS, which is designed to work with RAS output. The Boundary feature class defines the boundary of analysis for the floodplain.

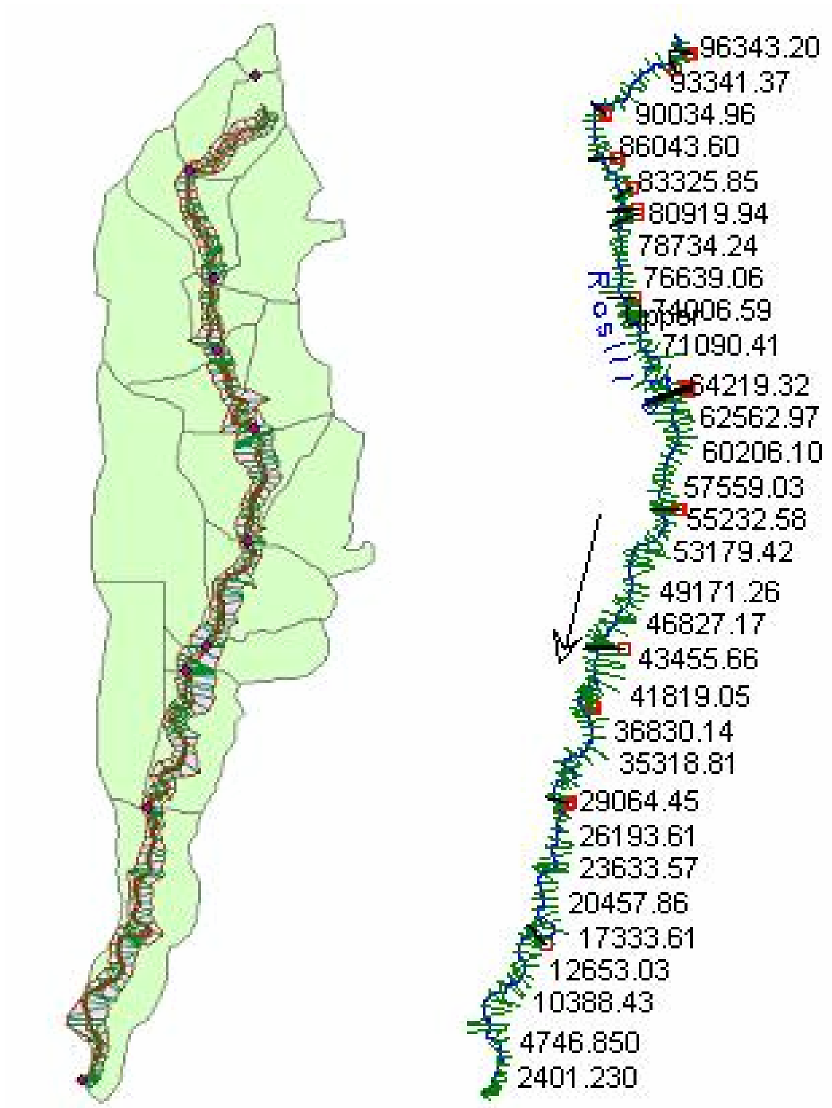


Figure 5.3 GEODATABASE (LEFT) AND RAS (RIGHT) VIEW OF CROSS SECTIONS

Both HMS and RAS use the HEC's Data Storage System (DSS) for storing time series information, so the Interface Data Models for HMS and RAS share a common DSSTSValues table and DSSTSType table. These tables extend the Arc Hydro TimeSeries and TSType tables to support parameters required by DSS.

Attributes of DSSTSValues

OBJECTID	DSSTSID	TSDateTime	TSValue
4502	3	1/21/1991	2680.237793
4503	3	1/21/1991	2648.336670
4504	3	1/21/1991	2611.453125
4505	3	1/21/1991	2560.596680
4506	3	1/21/1991	2486.648193
4507	3	1/21/1991	2384.590332
4508	3	1/21/1991	2257.026611

Record: 1 Show: All Selected Records

Figure 5.4 SAMPLE DSSTSVALUES TABLE

Attributes of DSSTSType

PathName	A_Part	B_Part	C_Part	D_Part	E_Part	F_Part
/Rosillo Creek/569154/PRECIP-I	Rosillo Creek	569154	PRECIP-INC	30JUN2002	1HOUR	NEXRAD
/Rosillo Creek/569155/PRECIP-I	Rosillo Creek	569155	PRECIP-INC	30JUN2002	1HOUR	NEXRAD
/Rosillo Creek/568155/PRECIP-I	Rosillo Creek	568155	PRECIP-INC	30JUN2002	1HOUR	NEXRAD
/Rosillo Creek/568154/PRECIP-I	Rosillo Creek	568154	PRECIP-INC	30JUN2002	1HOUR	NEXRAD
/Rosillo Creek/568153/PRECIP-I	Rosillo Creek	568153	PRECIP-INC	30JUN2002	1HOUR	NEXRAD

Record: 1 Show: All Selected Records (0 out of 15 Selected.) Options

Figure 5.5 SAMPLE DSSTSTYPE TABLE

Both the HMS and RAS Interface Data Models are incorporated into a single geodatabase for Rosillo Creek. Each Interface Data Model facilitates communication with its particular simulation model. Both Arc Hydro features (such as HydroJunctions) and the DSS time series tables serve to transfer information between Interface Data Models for each simulation model.

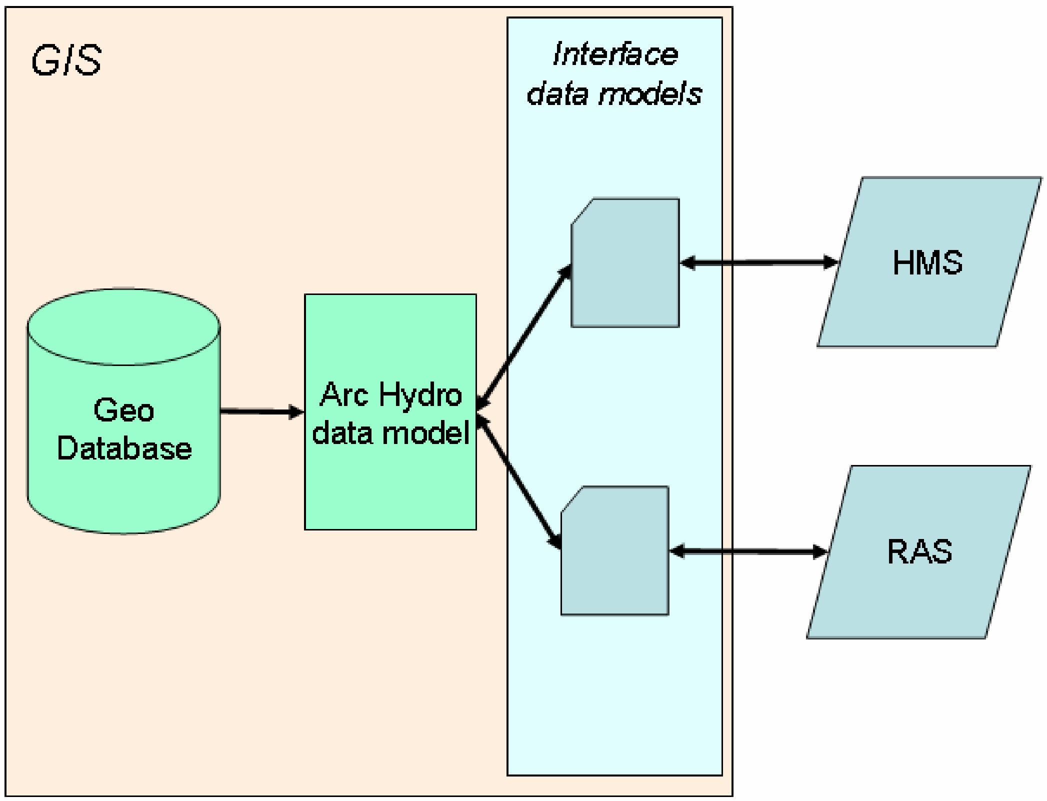


Figure 5.6 DATA MODEL FOR NEXRAD TO FLOOD MAP APPLICATION

5.2.3 Data Preparation

Before any simulations can be run, data must be prepared in the geodatabase, as well as the simulation files required by HMS and RAS. GIS data were obtained from the City of San Antonio, including the stream network, cross sections, watersheds, and 2-ft contours. The contours were used to generate a digital elevation model (DEM) for the area. Once the data were loaded into a geodatabase, the Arc Hydro schema was applied. Subsequently, the resulting geodatabase was further adapted to incorporate the HMS and RAS Interface Data Models. Information exchange points were established between the GIS and HMS at the outlets of 17 Watersheds within the basin. These outlets were stored in the HydroJunction feature class, and located on the stream network. The information exchange points between the GIS and RAS were established at each next downstream CrossSection from the HydroJunction serving as each Watershed's outlet. These CrossSections were also represented by a HydroJunction on the stream network. The configuration of 223 CrossSections in Rosillo Creek was considered dense enough so that a Watershed's hydrograph could be associated with the next downstream CrossSection without compromising the integrity of the data. All CrossSections were used when importing RAS results back into the geodatabase.

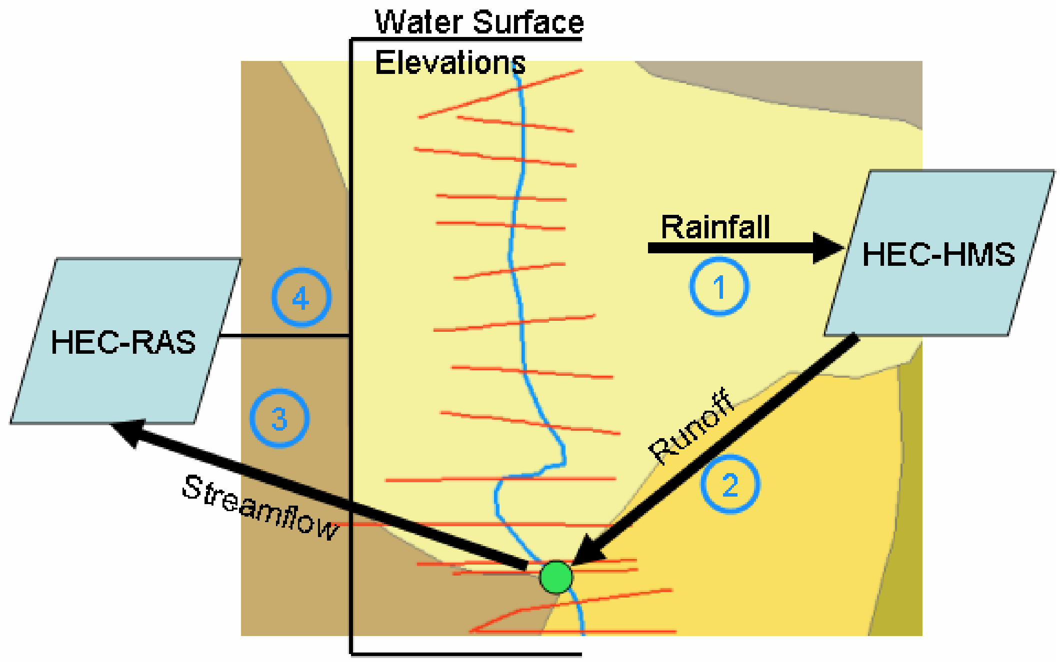


Figure 5.7 Time Series exchanged at Information exchange Points, including 1) Watersheds, 2) Watershed Outlet Junctions, and 3) Cross Sections, to produce 4) water surface elevations on all Cross Sections

HMS and RAS project files (and other supporting files) were also set up for Rosillo Creek. These files contain the information necessary to run an HMS or RAS simulation. Certain sections of those files reflect inputs from the GIS, such as rainfall data for an HMS meteorological record. Features in the HMS and RAS files possess identifiers to link them with features in the geodatabase, as described in the Data Model section above.

5.2.4 Model Description

The application in this case study was implemented by creating an ArcGIS 9 model called "NEXRAD to Flood Polygon", using ModelBuilder. The model contains 19 tools, including script tools, model tools, and standard ArcGIS tools. The script tools call both DLLs and executables to perform advanced work.

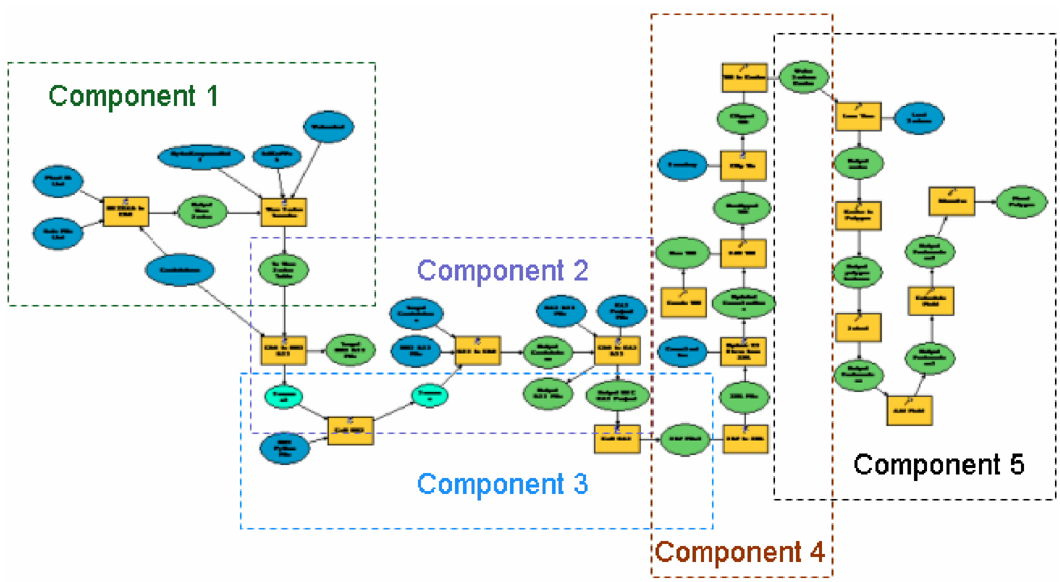


Figure 5.8 "NEXRAD TO FLOOD POLYGON" MODEL

The model is organized into five major components. The general order of execution for the model is from Component 1 to Component 5, with some overlapping execution between Component 2 and Component 3.

Component 1: NEXRAD to Watershed TimeSeries

Component 1 ingests NEXRAD rainfall data into the geodatabase for a set of NEXRAD polygons, and then transfers the rainfall time series from the NEXRAD polygons to Watersheds for the Rosillo Creek basin.

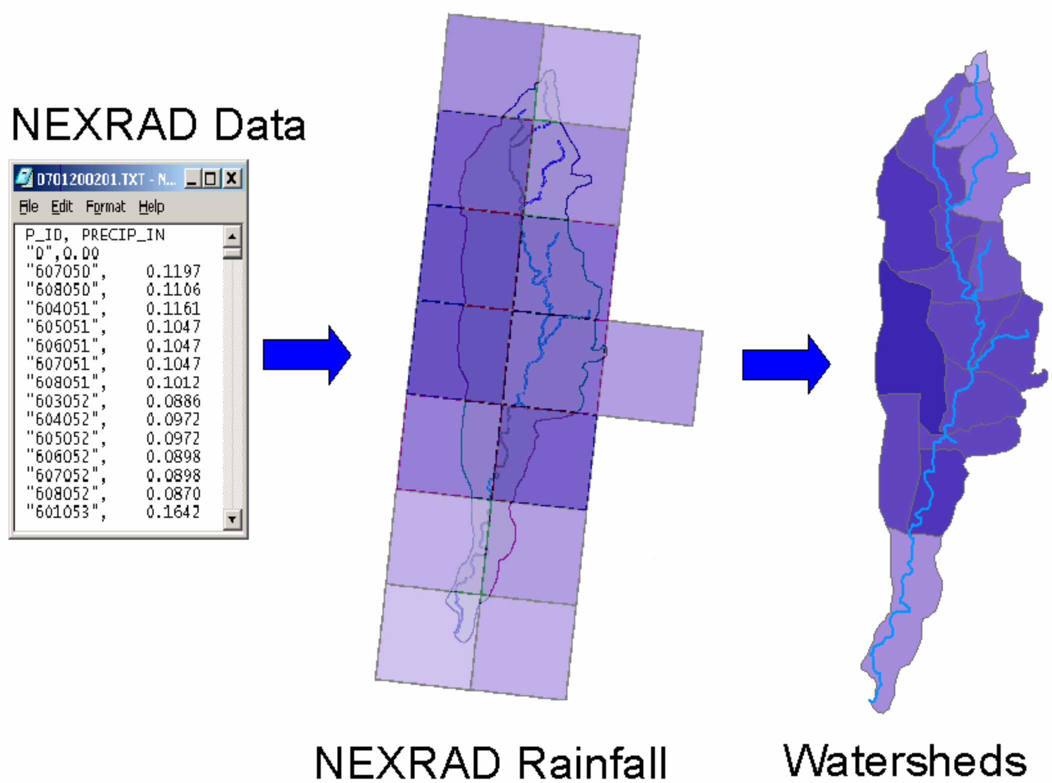


Figure 5.9 ASSOCIATING NEXRAD RAINFALL DATA WITH ROSILLO CREEK WATERSHEDS

NEXRAD to GDB

The NEXRAD to GDB script tool uses a PixelID text file to identify NEXRAD cells for which time series data are available. The data are stored in individual ASCII data files, which are indexed by a DataFileList text file. The tool matches available cells with NEXRAD polygon features in the geodatabase, and then imports the time series of rainfall into the Arc Hydro TimeSeries table.

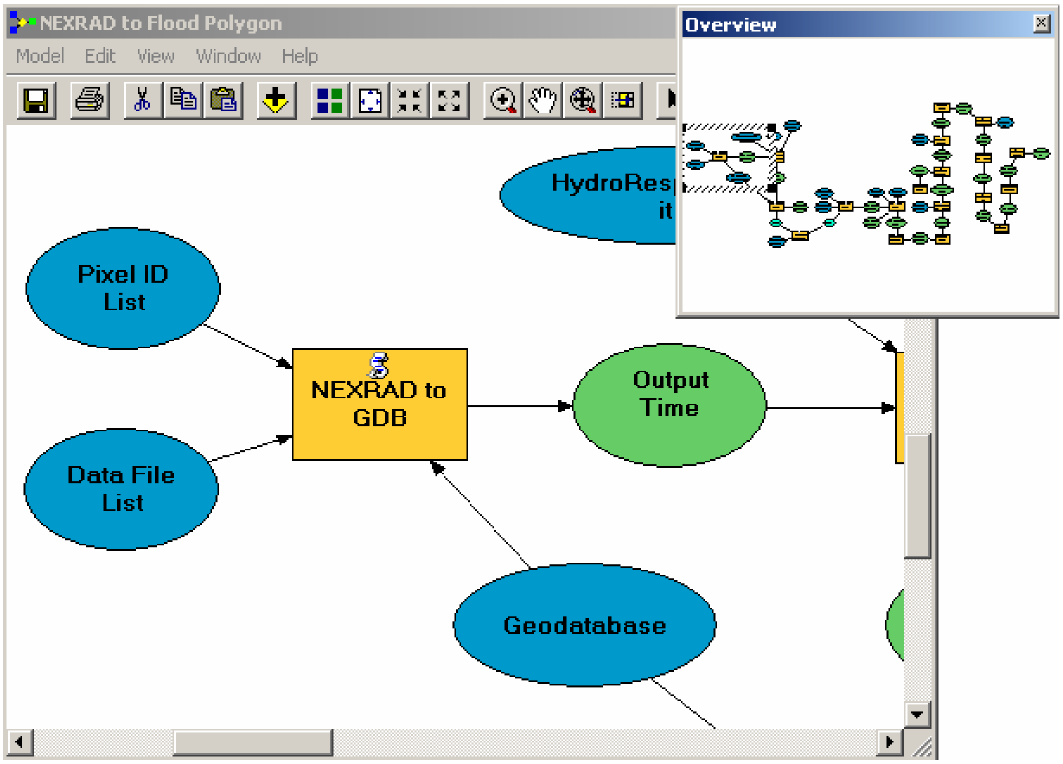


Figure 5.10 NEXRAD TO GDB SCRIPT TOOL

Time Series Transfer

The Time Series Transfer script tool transfers time series associated with the NEXRAD polygons to Watersheds for the Rosillo Creek basin. The tool analyzes the extent to which each NEXRAD cell covers a given Watershed, and then based on that extent, calculates a weighted average of rainfall for that Watershed at each time step.

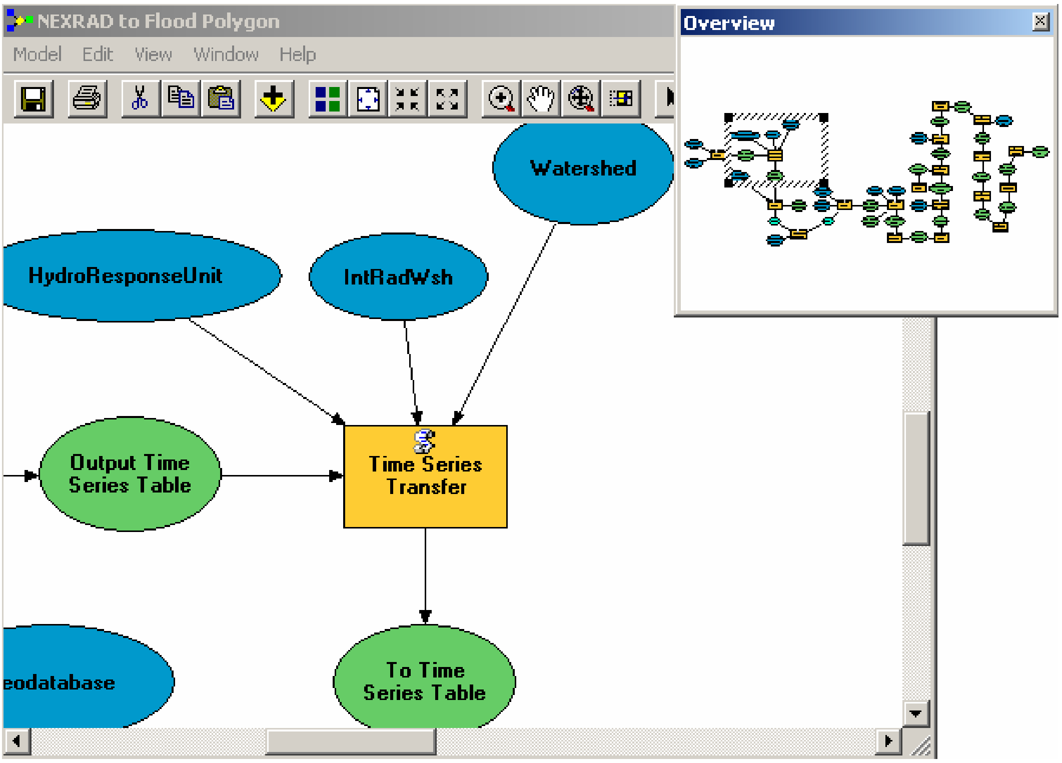


Figure 5.11 **TIME SERIES TRANSFER SCRIPT TOOL**

Component 2: Geodatabase/DSS Time Series Transfer

Component 2 contains bridges to transfer time series between the geodatabase and HEC's DSS time series format.

GDB to HMS DSS

The GDB to HMS DSS script tool transfers rainfall time series data associated with each Watershed from the geodatabase to an HMS DSS file. Each Watershed in the geodatabase has a corresponding basin object in the HMS basin file, identified by the HMSCode attribute. This tool is called prior to calling

HMS, so that the HMS project will have access to the latest rainfall data from the GIS.

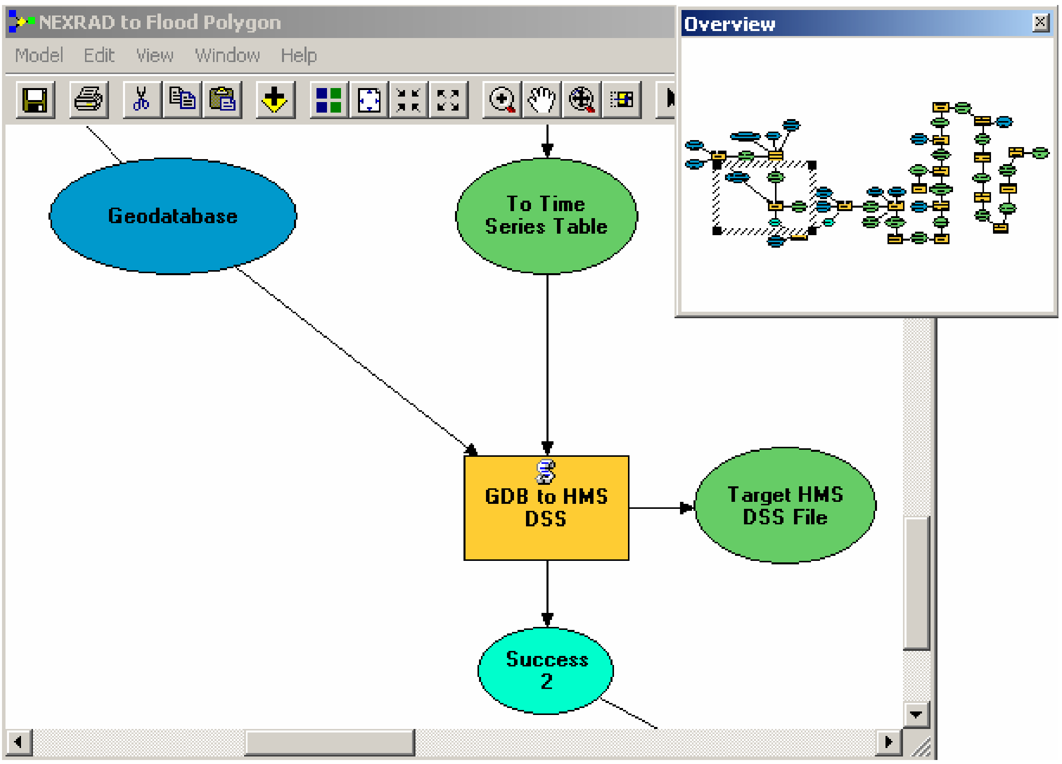


Figure 5.12 GDB TO HMS DSS SCRIPT TOOL

DSS to GDB

The DSS to GDB script tool transfers time series data from an HEC DSS file to a geodatabase time series table. This tool is run after HMS has completed its simulation, in order to bring runoff hydrograph time series into the geodatabase. The time series are associated with HMS nodes in the geodatabase and are identified by an HMSCode attribute. These nodes are stored in the SchemaNode feature class, and are related to HydroJunctions at the appropriate location on the stream network.

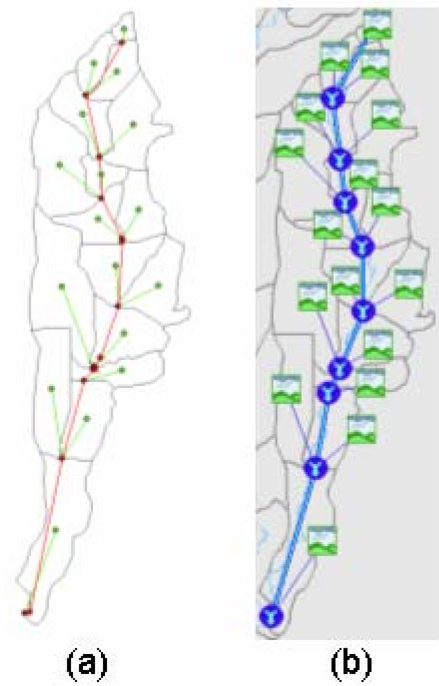


Figure 5.13 (A) ARC HYDRO SCHEMATIC NETWORK, AND (B) CORRESPONDING HMS SCHEMATIC

CrossSections describing the river channel are also related to HydroJunctions on the stream network. Through HydroJunction relationships, the next downstream CrossSection from a given SchemaNode can be located. Therefore, each runoff hydrograph time series can be matched to the cross section closest to the node where that streamflow occurs in the network.

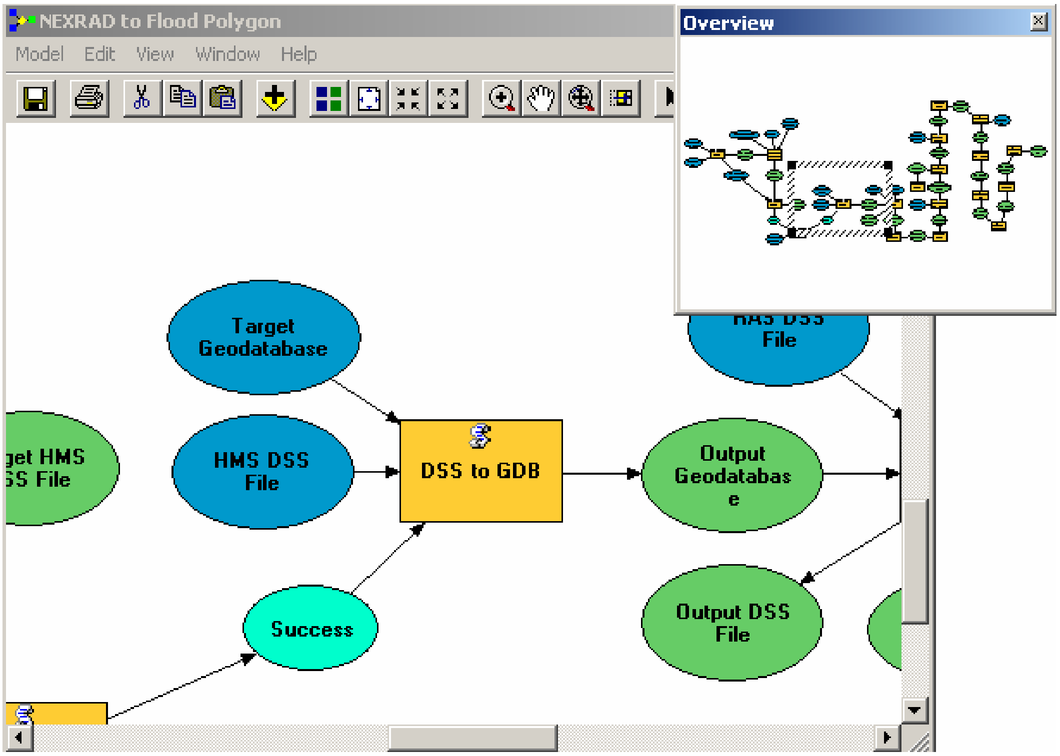


Figure 5.14 DSS TO GDB SCRIPT TOOL

GDB to RAS DSS

The GDB to RAS DSS script tool transfers time series data from the geodatabase to a RAS DSS file. Additionally, the tool updates the RAS project file to reflect the new time series records. This tool is run prior to RAS execution, to provide streamflows at certain cross sections.

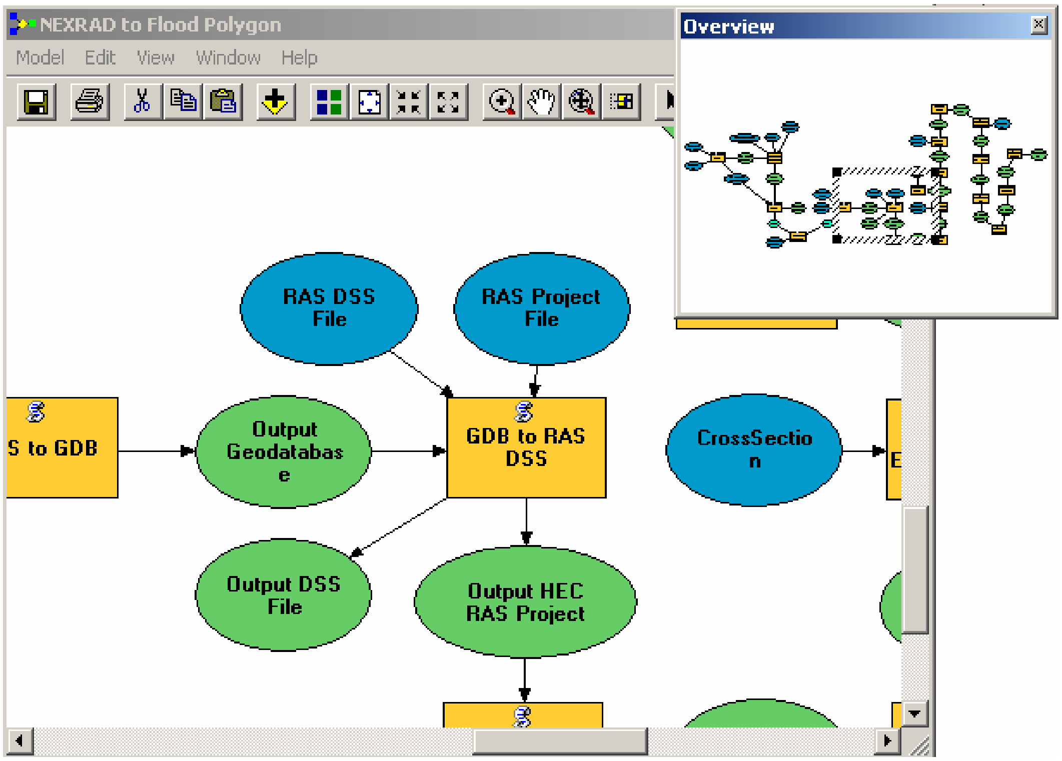


Figure 5.15 GDB to RAS DSS Script Tool

Component 3: Simulation Model Execution

Component 3 calls simulation models using script tools. Once the necessary DSS files have been updated by other components in the ArcGIS model, the simulation models run independently of other processes. The ArcGIS model waits until a given simulation model has finished before continuing with the next tool in the sequence.

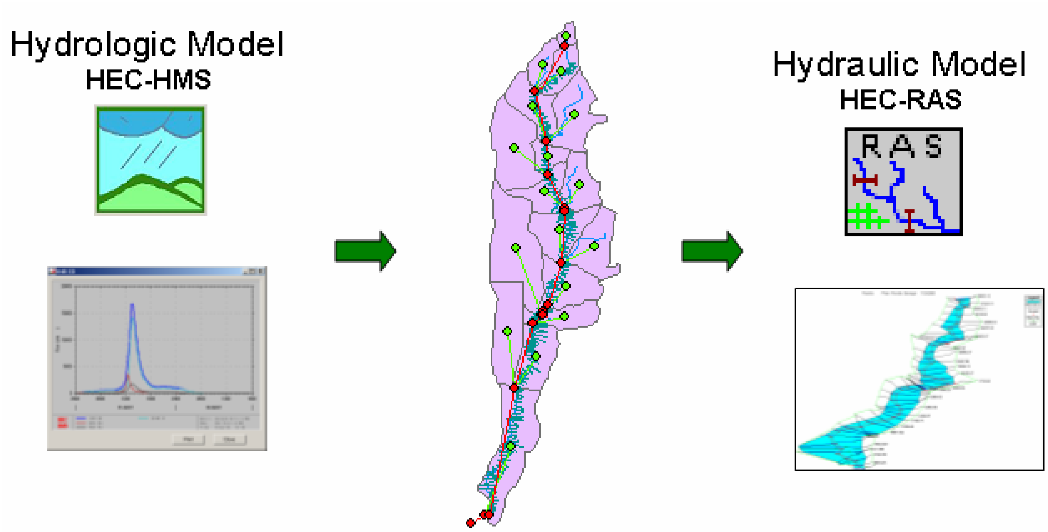


Figure 5.16 SIMULATION MODELS INTEGRATED THROUGH THE GEODATABASE

Call HMS

The Call HMS script tool calls HEC-HMS to perform rainfall-runoff calculations for each Watershed in the Rosillo Creek basin. The input rainfall time series are obtained for each Watershed from the geodatabase using the GDB to HMS DSS tool. The result is a set of runoff hydrographs for nodes in the stream network, with the nodes representing Watershed outlets and stream confluences.

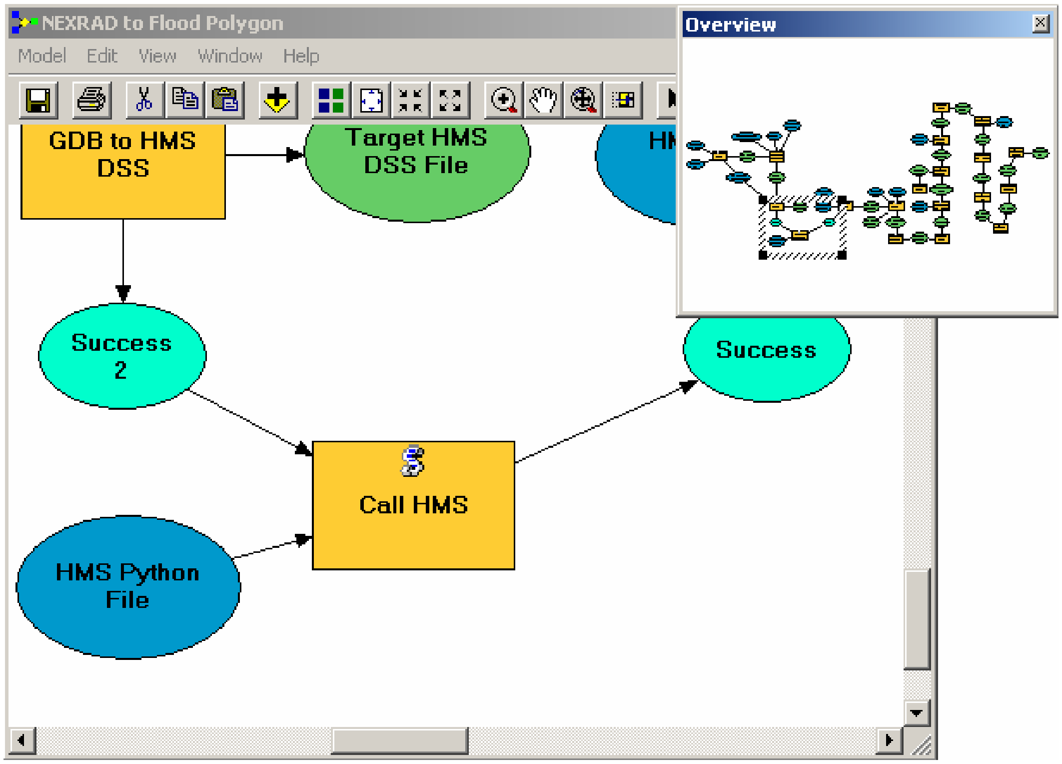


Figure 5.17 CALL HMS SCRIPT TOOL

Call RAS

The Call RAS script tool calls HEC-RAS to calculate the water surface elevation at each Cross Section. The input runoff time series are obtained for a set of Cross Sections from the geodatabase using the GDB to RAS DSS tool.

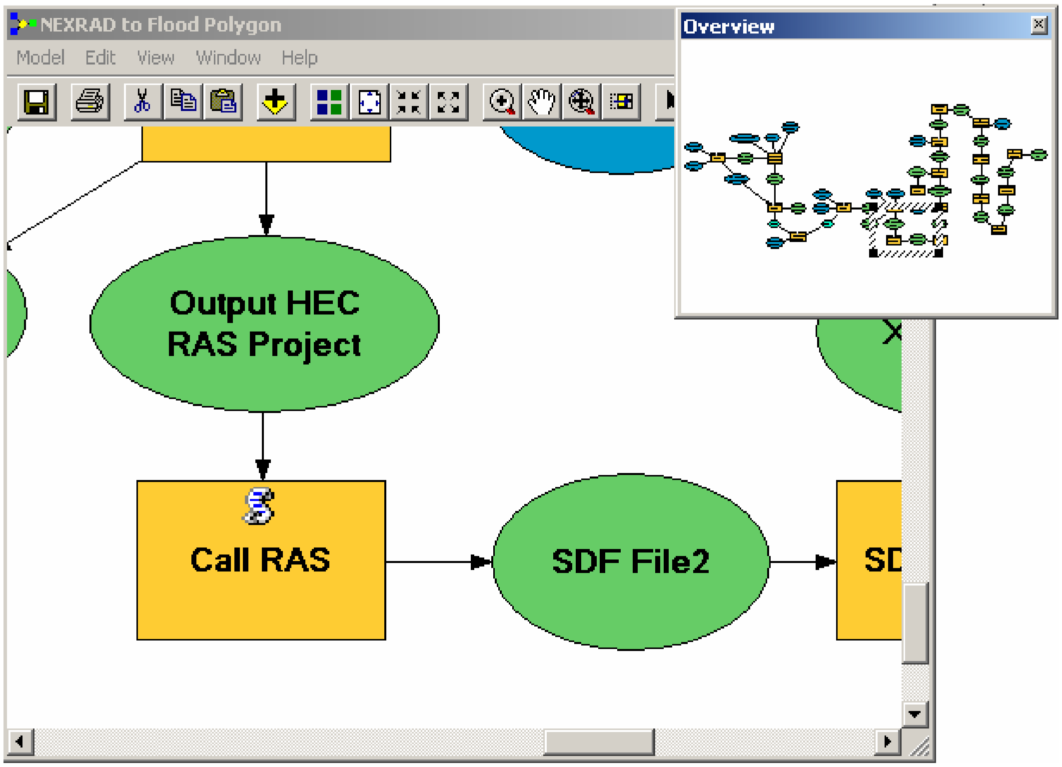


Figure 5.18 CALL RAS SCRIPT TOOL

Component 4: Water Surface Raster Generation

Component 4 uses the output from a RAS simulation to generate a raster representing water surface elevations in the GIS.

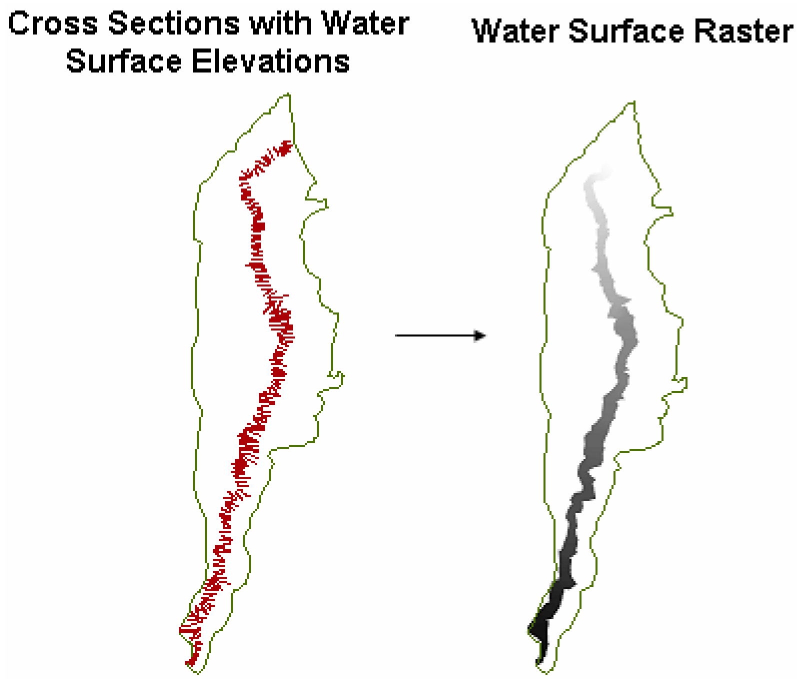


Figure 5.19 WATER SURFACE ELEVATIONS ON CROSS SECTIONS ARE USED TO GENERATE A WATER SURFACE RASTER

Updating Cross Sections from RAS Output

Two tools are used to update cross sections in the geodatabase from RAS output. An SDF file is the output file from a RAS simulation, which contains water surface elevations for each cross section in the RAS model. The SDF to XML script tool converts the SDF file to XML format, which is an easier format to import into the geodatabase. The Update XS Elevs from XML script tool uses that XML file to update the attribute in the CrossSection feature class that describes water surface elevation.

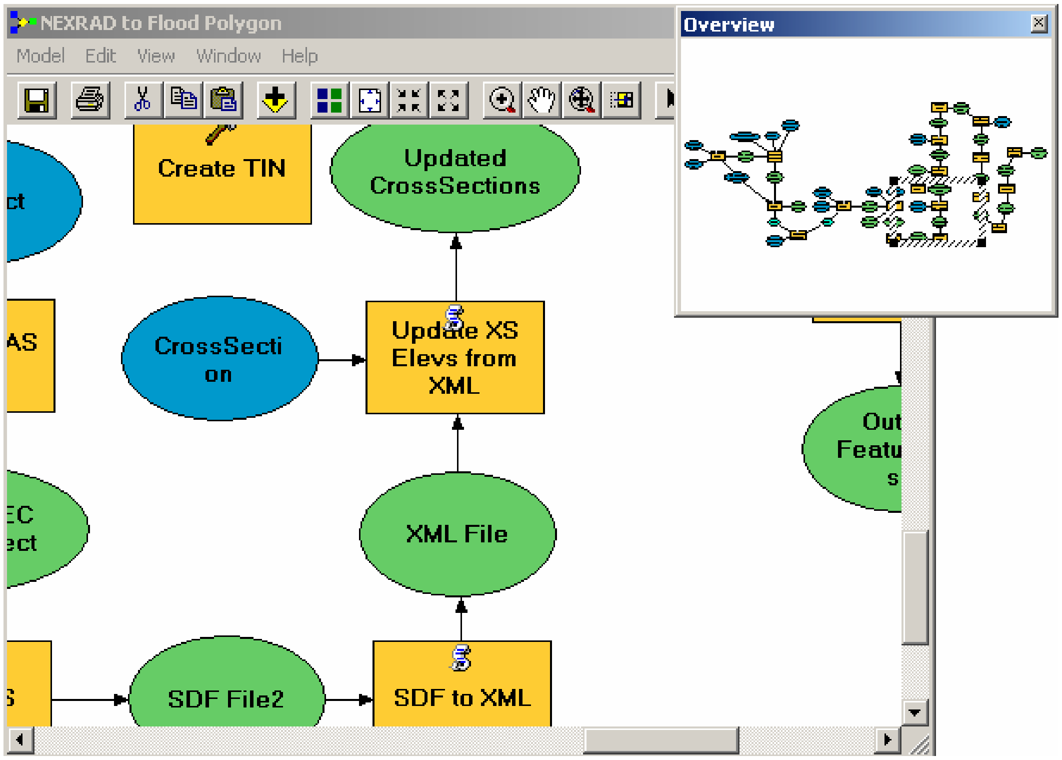


Figure 5.20 UPDATING CROSS SECTIONS FROM RAS OUTPUT

Creating the Water Surface TIN

Once water surface elevations have been attributed on CrossSections, three tools are used to create a TIN representing water surface elevations. The Create TIN tool initializes a new TIN. The Edit TIN tool uses the CrossSections as soft break lines with elevations taken from the water surface elevation attribute. The Clip TIN script tool clips the resulting TIN to the analysis boundary defined by the Boundary feature class.

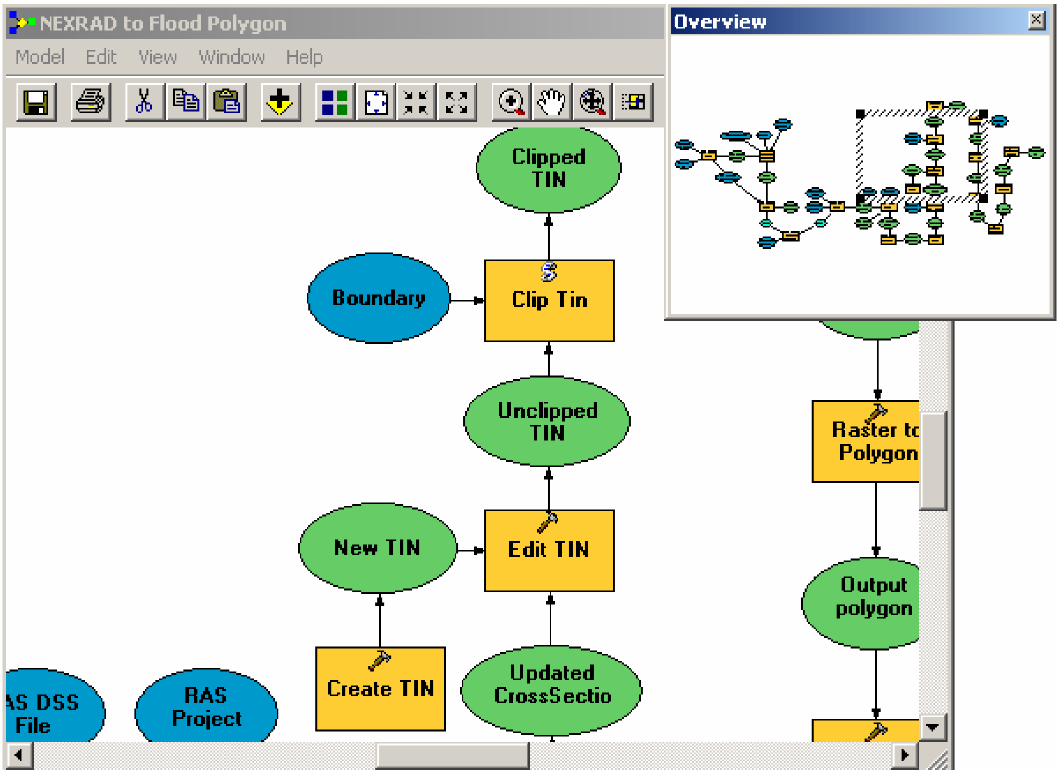


Figure 5.21 CREATING THE WATER SURFACE TIN

TIN to Raster

The TIN to Raster tool converts the water surface elevation TIN to a raster for further analysis.

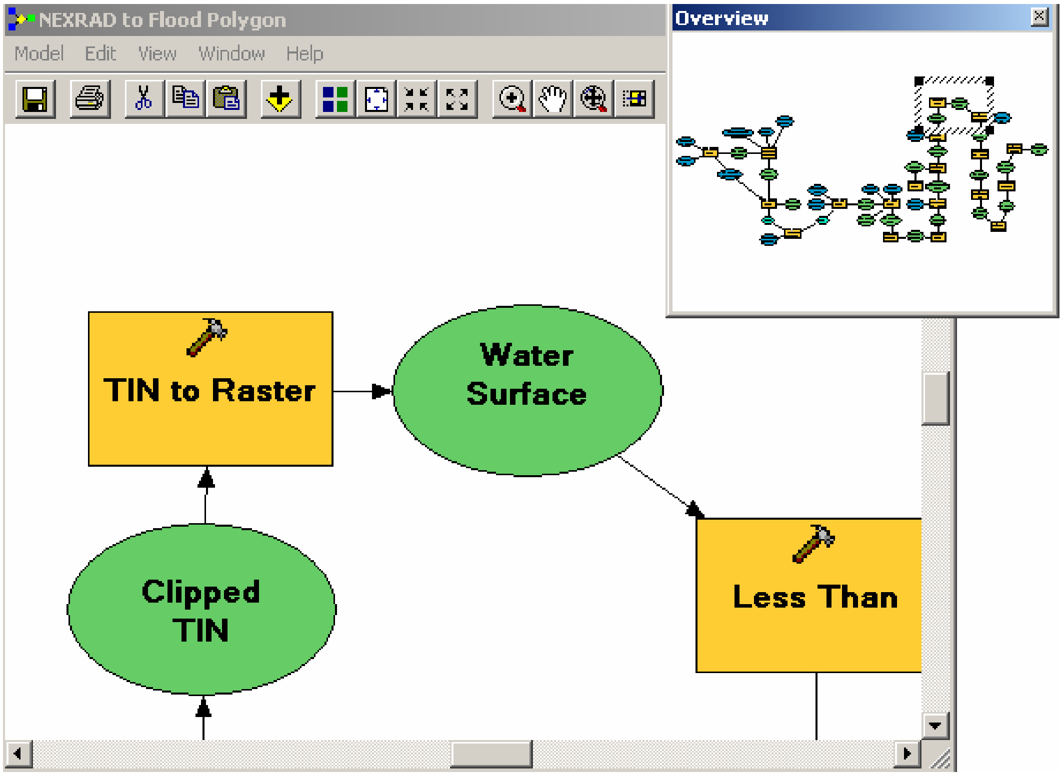


Figure 5.22 TIN TO RASTER TOOL

Component 5: Flood Polygon Generation

Component 5 uses the water surface elevation raster and a raster representing the land surface to create a polygon representing the area inundated by flood waters.

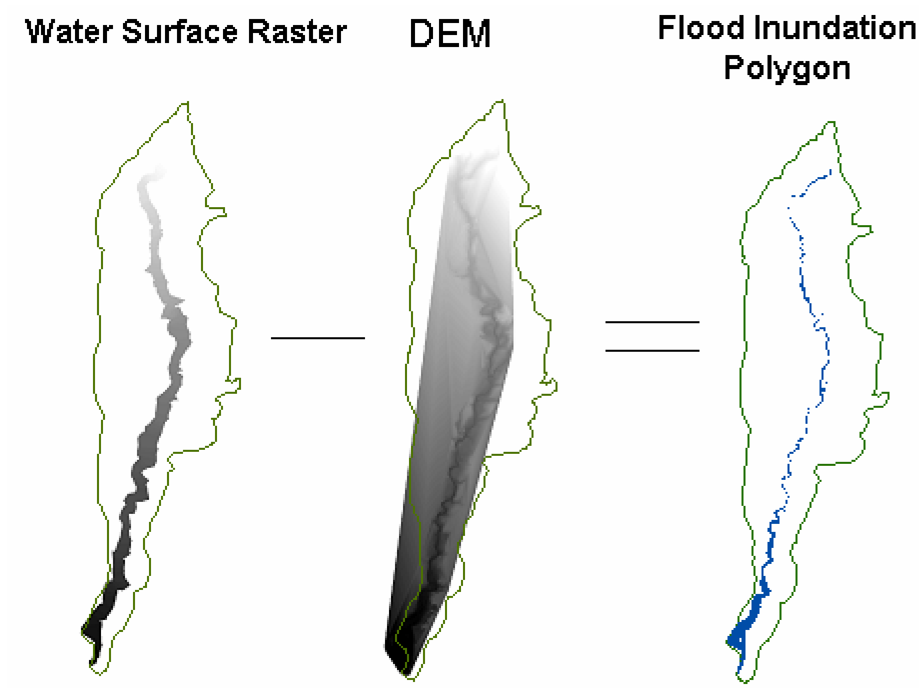


Figure 5.23 FLOOD INUNDATION POLYGON IS PRODUCED BY SUBTRACTING THE LAND SURFACE FROM THE WATER SURFACE

Creating the Flood Polygon

Two tools are used to convert raster surface data to raw polygons, which will be used to create the final inundation polygon. The Less Than tool subtracts the land surface elevation raster from the water surface elevation raster to create a raster representing the depth of inundation. The Raster to Polygon tool converts the resulting raster to a set of polygon features. One feature is created for each depth value from the raster. Polygons representing flooded areas have a positive value, while those representing dry areas have a negative value.

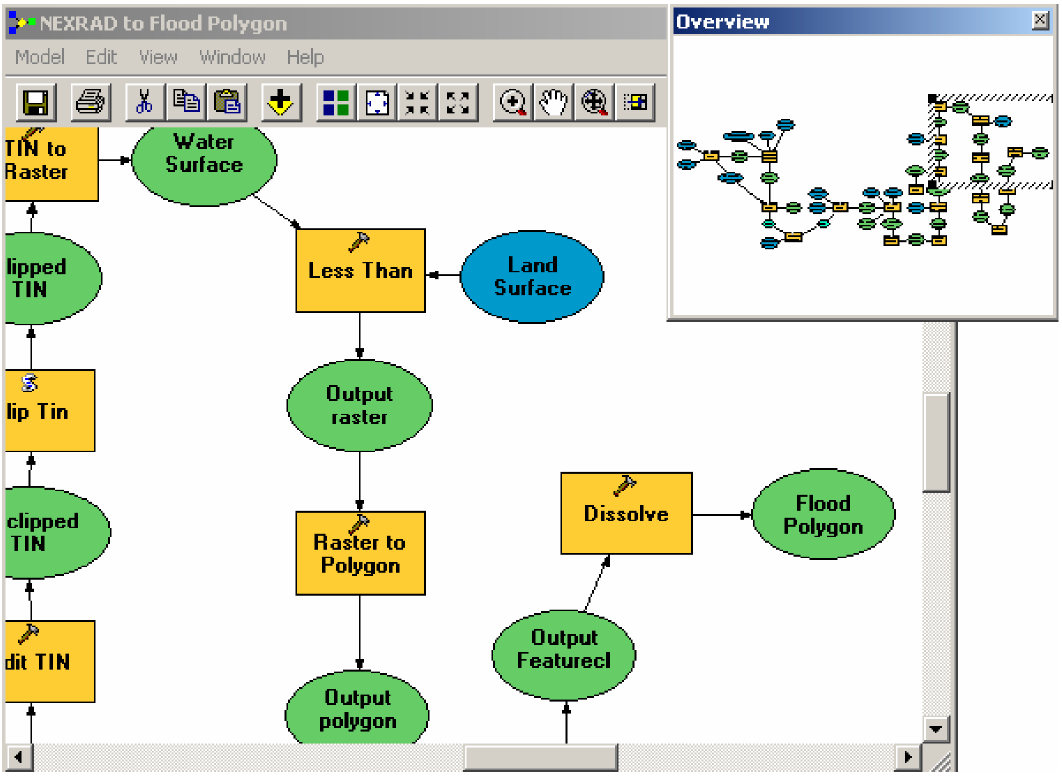


Figure 5.24 CREATING THE FLOOD POLYGON

Isolating Ponded Polygons

Three tools are used to isolate polygons representing inundated areas. The Select tool selects and creates a new feature class from those flood polygons with a value greater than zero. These polygons represent inundated areas. The Add Field tool adds a short integer field called Inundated to the new polygon feature class. The Calculate Field tool calculates a value of 1 for all features in the new feature class. Now, all polygons representing inundated areas are captured in their own feature class, and have a common identifier of 1.

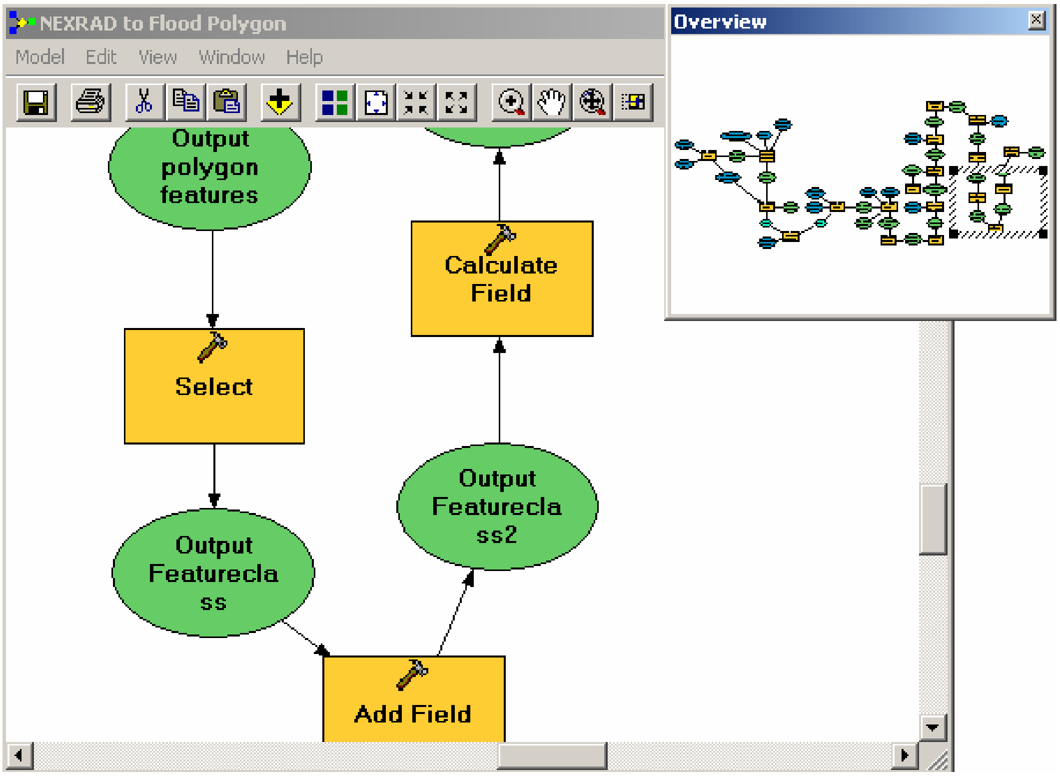


Figure 5.25 ISOLATING PONDED POLYGONS

Dissolve

The Dissolve tool is the last tool in the model. This tool dissolves all of the ponded polygons based on the Inundated attribute of 1, to produce a single feature representing inundated area. This feature is stored in the InundatedArea feature class in the geodatabase.

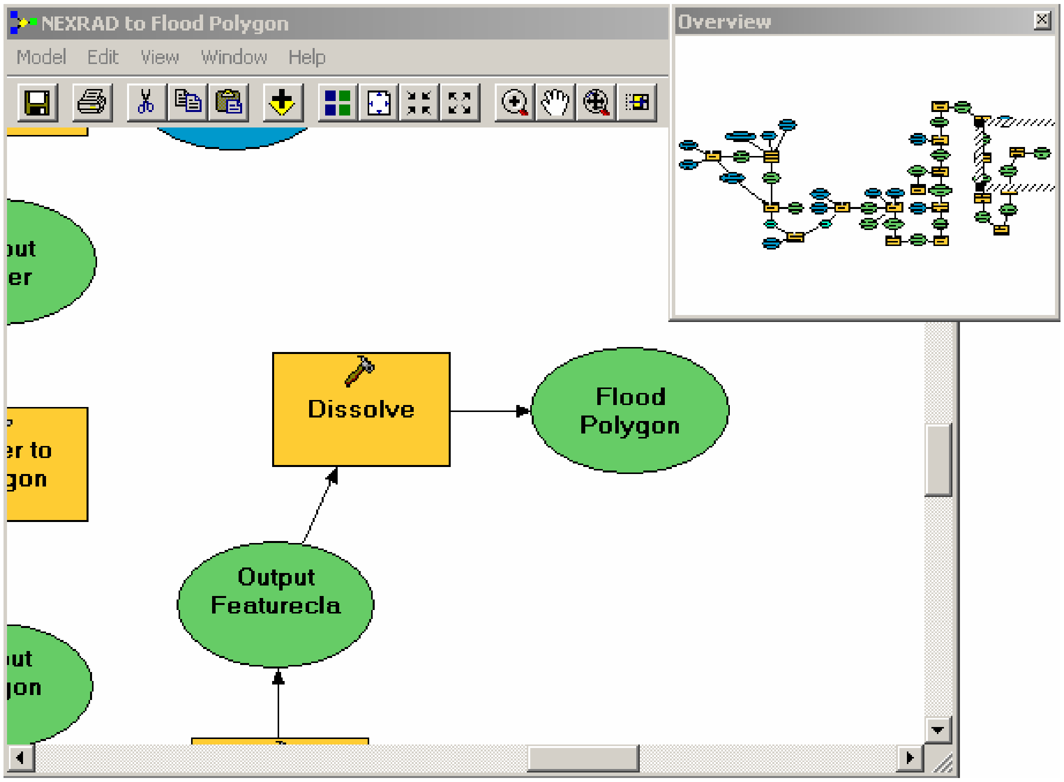


Figure 5.26 DISSOLVE TOOL

5.2.5 Results

The result from execution of the NEXRAD to Flood Polygon model is a polygon representing the area inundated by flood waters from a given storm. Different time series of rainfall may be substituted in the model by simply changing the data file list parameter in the NEXRAD to GDB tool. Other components of the model may also be changed, such as the cross sections used or even the basin itself, by simply dragging and dropping different datasets onto the model diagram.

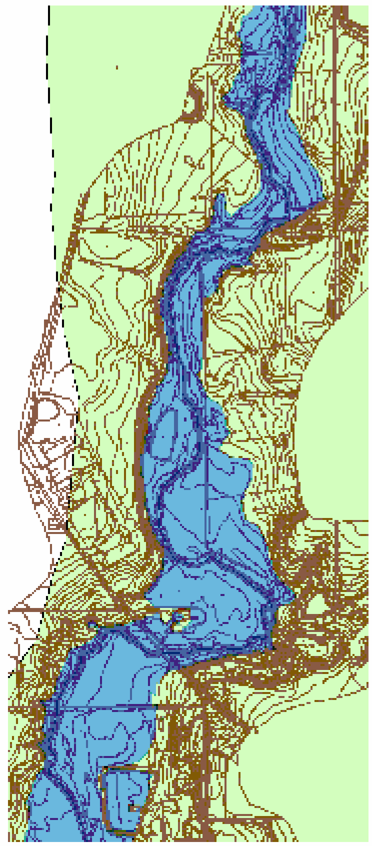


Figure 5.27 POLYGON OF INUNDATED AREA PRODUCED BY NEXRAD TO FLOOD POLYGON TOOL

5.2.6 Future Work

Currently, the model only generates a single polygon per execution, representing the steady state solution. A future enhancement would include the ability to produce multiple polygons for unsteady flow, so that an animation could be produced to show inundated area over time. Smaller enhancements include better error handling and a more robust tool design to facilitate the extension of model components towards other applications.

The NEXRAD to Flood Polygon tool will provide a valuable tool in evaluating management alternatives for flood control plans, land development, and flood scenarios for the San Antonio River Authority, the City of San Antonio, and Bexar County. Additionally, the tool serves as a valuable prototype in linking GIS and simulation models through Arc Hydro and the ArcGIS ModelBuilder environment.

Chapter 6 Integration of Features with Processing Engines

Arc Hydro provides a framework for organizing and preprocessing geospatial and temporal data for use in hydrologic and hydraulic simulation models. One of the products created by Arc Hydro is the *schematic network*, which is a graphical representation of connectivity between features in the landscape through a network of nodes and links. Each node represents a given feature, while each link connects two nodes, showing connectivity between the features represented by those nodes. This chapter describes a methodology for processing schematic networks using ArcGIS 9.0 geoprocessing technology, followed by example applications of how the schematic processor is used to integrate features with *processing engines* to simulate hydrologic behavior. A processing engine is a custom DLL which defines how a given type of schematic feature communicates information to other schematic features.

6.1 Arc Hydro Schematic Network

Through the attributes and relationships of Arc Hydro, a schematic network may be created that describes the connectivity between features. A schematic network may be created automatically from up to two Arc Hydro feature classes, although more feature classes may be included manually. Typically, Watersheds and HydroJunctions are used to create the schematic network. The JunctionID attribute on a Watershed stores the HydroID of the HydroJunction that serves as the outlet for that Watershed. The NextDownID attribute of a HydroJunction stores the HydroID of the next downstream

HydroJunction. Figure 6.1 shows JunctionIDs of three Watersheds matching up with the HydroIDs of each Watershed's outlet HydroJunction.

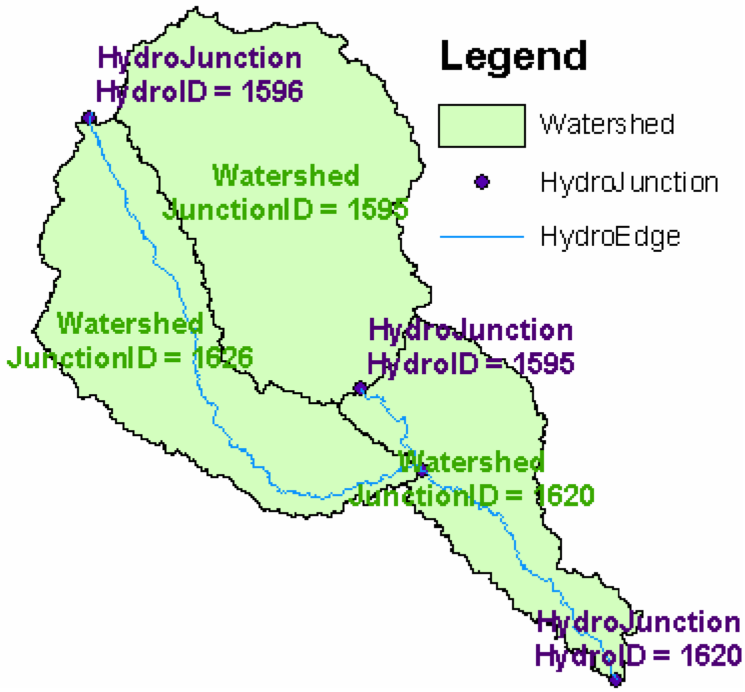


Figure 6.1 Geospatial Data Representing Watersheds and the Hydro Network

With these attributes, connectivity is established between Watersheds and HydroJunctions. From this connectivity, a schematic network representing those features may be created.

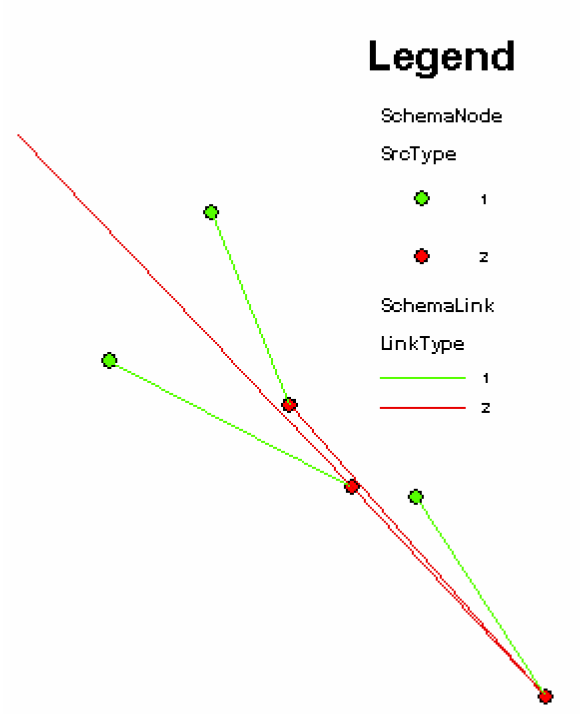


Figure 6.2 Schematic Network Representing Watersheds and HydroJunctions

The schematic network is composed of two feature classes: SchemaNode and SchemaLink, representing schematic nodes and schematic links, respectively. A schematic node represents a hydrologic feature, such as a watershed or a junction in the stream network. Schematic nodes are typically created from polygon or point features, with a node being located at the centroid of its related polygon feature or at the same location as its related point feature. The schematic nodes contain a field called FeatureID, which stores the HydroID of the source feature that is represented by the schematic feature.

Schematic links are straight lines that connect Schematic nodes. A schematic link cannot be connected to more than two schematic nodes, although a schematic node could be connected to several schematic links. In other words, a schematic link exists for every connection that a schematic node has with other schematic nodes. Schematic links contain FromNodeID and ToNodeID, which identify the HydroID of the schematic nodes connected to a given schematic link. FromNodeID identifies the from node, and ToNodeID identifies the to node.

The schematic features are further subdivided into two types. Type 1 features are associated with a JunctionID-HydroID relationship from the source features, such as JunctionID on Watershed pointing to the HydroID of that Watershed's outlet HydroJunction. Type 2 features are associated with a NextDownID-HydroID relationship from the source features (see Fig. 6.3). The types are specified in the SrcType attribute of SchemaNode, and the LinkType attribute of SchemaLink. Additional node or link types may be added to denote additional feature categories.

Once the schematic network has been created, attributes from the source features may be copied to the schematic features using the FeatureID-HydroID association between a schematic node and its related Arc Hydro feature. These attributes are stored in fields, which are supplemental to the base schematic network attributes defined by Arc Hydro. Additional attributes to further describe schematic features may also be added.

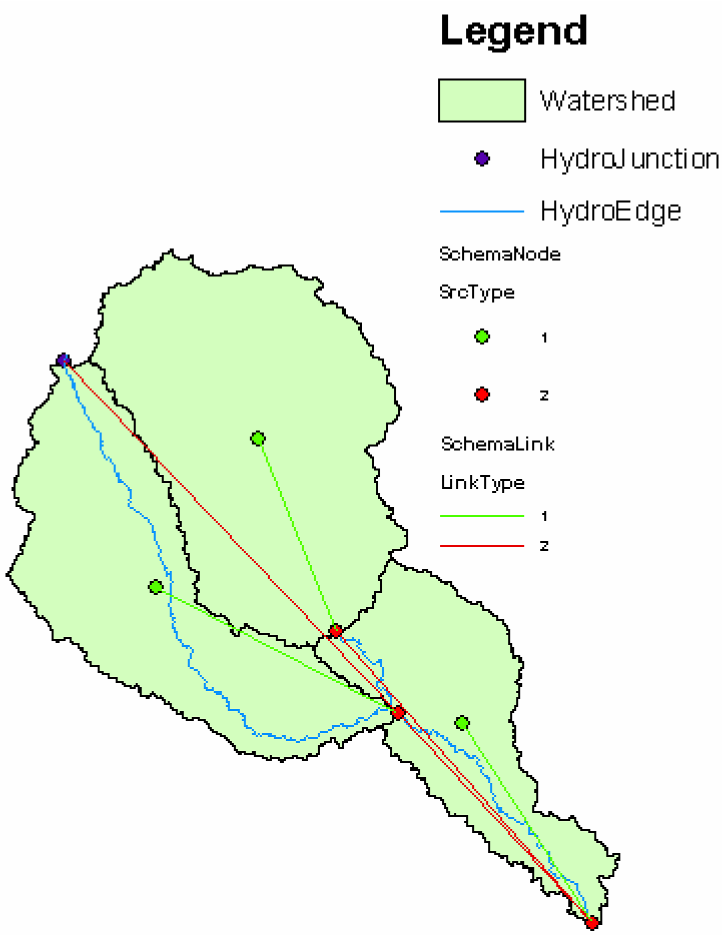


Figure 6.3 Schematic Network Overlain on Watersheds and the Hydro Network

6.2 Methodology

The schematic processor extends the functionality of the Arc Hydro schematic network to include behaviors that process attributes throughout the network. Each feature in a schematic network has a set of attributes that describe the state of that feature and what information it will pass to the next downstream feature in the network. Also associated with each feature is a set of behaviors that define how the feature responds to information from upstream features, and how the feature determines what information will be passed to the next downstream feature.

This methodology makes the following assumptions:

- The schematic network is dendritic. If branching occurs, information from the node at a branch will be passed to all downstream links at the branch.
- No backwater effects occur. The processing takes place from upstream to downstream features. Once an upstream feature has been processed, its values cannot be changed due to influences from downstream features.

Note that the user can create a workaround for each assumption by implementing custom processing engines, which are described later in this chapter.

6.2.1 Schematic Values

The movement of water (or contaminants in the water) across the landscape can be simulated by passing values down through the schematic network. Schematic network features incorporate four types of values: **received** values, **incremental** values, **total** values, and **passed** values.

Value Type	Definition
Received	Values received from upstream features
Incremental	Value contributed directly by the current schematic feature
Total	Combination of Received and Incremental values for a given feature
Passed	Value that a given feature passes to the next downstream feature

Table 6.1 Definitions of Schematic Value Types

Received Values

Received values are those values received by a schematic feature from adjacent upstream schematic features, after the values passed from those upstream features have been processed by the receiving feature. A node can only receive values from adjacent upstream links, and a link can only receive a value from its adjacent upstream node.

As an example, consider the simple schematic network shown in Figure 6.4. Node 2 represents the most downstream feature in the network. Links 1 and 2 are connected to upstream schematic features beyond the dashed lines. Suppose that streamflow is being routed throughout the network. Link 1 and Link 2 both flow into Node 1. Therefore, the received values for Node 1 consist of the streamflows discharging from Link 1 and Link 2.

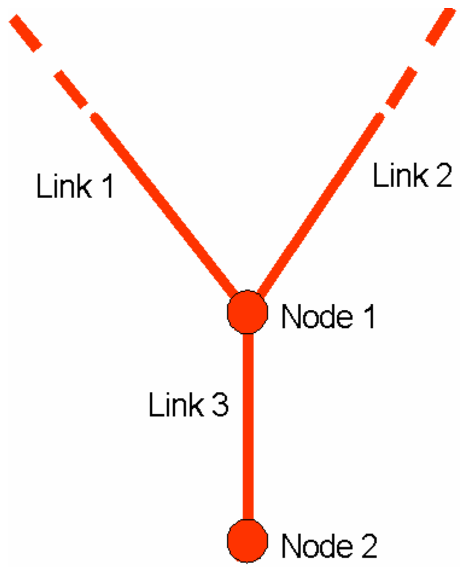


Figure 6.4 Sample Schematic Network

Once the streamflows are combined, Node 1 passes those flows to Link 3. Thus, the received value for Link 3 consists of the streamflow from Node 1.

Incremental Values

An incremental value is the value incorporated into the schematic network at a given schematic feature's location. Continuing the example above, suppose that Node 1 is also at the location of a rice field. A certain volume of water is withdrawn at that location for irrigation purposes. This water is removed from the schematic network. Thus Node 1 would possess a negative incremental value, which is taken into account when determining how much water from Links 1 and 2 is passed to Link 3.

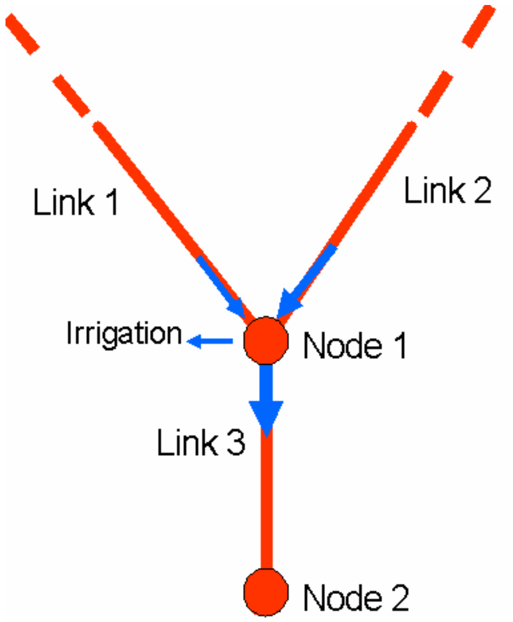


Figure 6.5 Water Withdrawal at Node 1 Represented as a Negative Incremental Value

Incremental values may also be positive. For example, at a node representing a watershed, the incremental value for that node could be the runoff produced in that watershed from a given rainfall event.

Total Values

The total value for a schematic feature is obtained by combining the incremental value and the received values, most commonly by simple addition. For example, the total flow for Node 1 in the above example is the sum of the streamflows from Links 1 and 2, minus the flow withdrawn for irrigation purposes.

Passed Values

A passed value is the value that a schematic feature passes to the next downstream feature in the network. Continuing the example above, suppose that the streamflow passed from Node 1 to Link 3 includes a bacterial load. As the bacteria travel along the length of the stream connecting the junctions represented by Nodes 1 and 2, suppose that the bacteria decay according to a first order decay rate. The bacterial load at the most downstream point of Link 3, after the decay rate has been applied, is the load that Link 3 passes to Node 2. Thus, Link 3 receives a certain value from Node 1, but passes a smaller value to Node 2 due to decay.

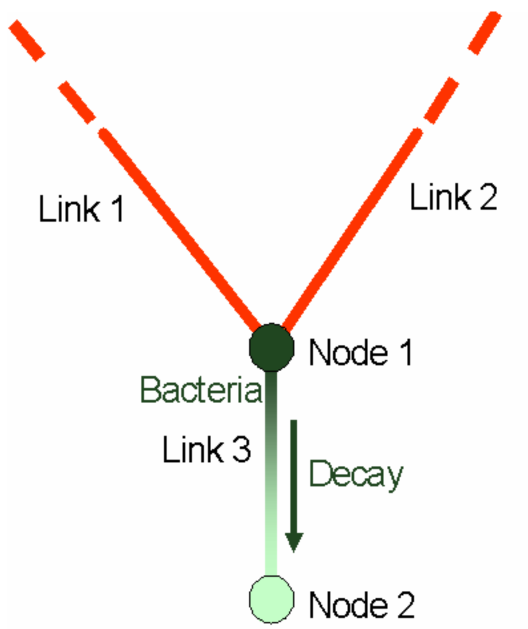


Figure 6.6 Passing of Bacterial Load from Link 3 to Node 2

6.2.2 Schematic Behaviors

When communicating with other schematic features, a given schematic feature possesses certain behaviors that control how that feature will accept values from upstream features, and how it will pass values to downstream features. These behaviors are called **receiving behavior** and **passing behavior**, respectively. Each behavior uses the values associated with a given schematic feature, along with other attributes and algorithms as necessary, to determine the final value for a given attribute of that feature.

Receiving Behavior

Receiving behavior defines how a schematic feature processes incoming values along with its incremental value to produce a total value for that feature. The simplest example of receiving behavior is to calculate the sum of all incoming values and the incremental value for a given feature. Using the example above, if Link 1 and Link 2 pass bacterial loads of 7 and 8 lb/day, respectively, and Node 1 has an incremental value of 2 lb/day, then the total value for Node 1 is simply the sum of those values, or 17 lb/day.

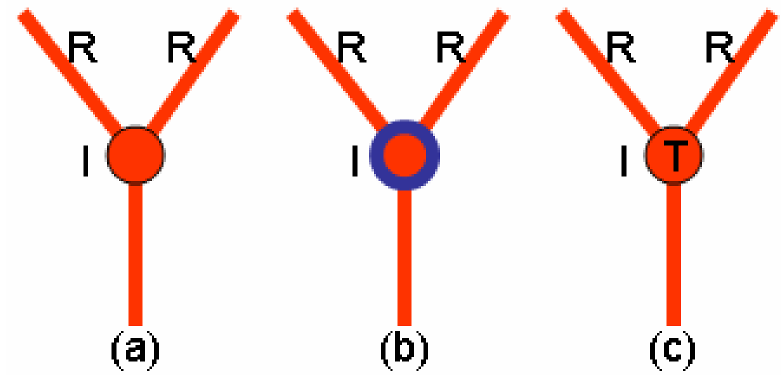


Figure 6.7 (a) Node in schematic network, (b) *Receiving* behavior called to combine *Received* values [R] and *Incremental* values [I], (c) *Total* value [T] produced from *Receiving* behavior

For a more complex example of receiving behavior, consider two links that pass a time series of information (rather than a single value) to a node. The node must combine the two time series to produce a total time series. This process involves grouping together and processing values based on each time step, which may require temporal interpolation of one or both time periods.

Passing Behavior

Passing behavior defines how a schematic feature processes its total value to produce a value that will be passed to the next downstream schematic feature. The simplest example of passing behavior is to pass the total value for a given feature.

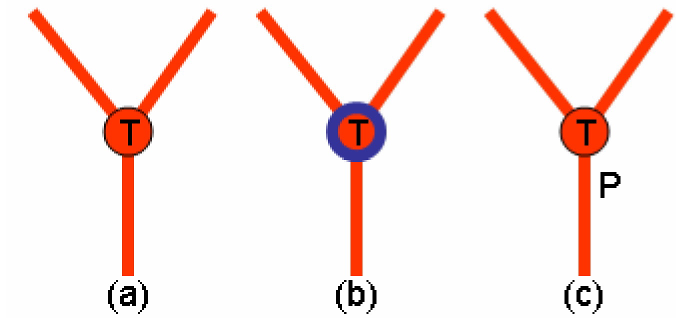


Figure 6.8 (a) Node in schematic network, (b) *Passing* behavior called to compute *Passed* value [P] from *Total* value [T], (c) *Passed* value produced from *Passing* behavior

More complex passing behaviors may also be modeled. Using the example above, the passing behavior for Link 3 involves a first order decay process. Thus, without any new bacterial load inputs, the load passed to Node 2 from Link 3 will be less than the load received by Link 3, due to decay along the length of the link. Note that the schematic processor does not define *where* the passing behavior takes place on the link (e.g., at the beginning, middle, or end, or along the entire length of the link). It merely defines how the total value for the link is changed to produce a value that is passed to its downstream node.

6.2.3 Processing Order

This methodology applies to schematic networks in which no backwater effects occur. In other words, each schematic feature receives no information from or about downstream schematic features. However, since each feature does receive information from upstream features, all upstream features should be processed (so that they know what value they should pass) before processing a given schematic feature.

Thus a processing order must be established so that upstream features are processed before downstream features. An implementation of an ordering scheme is described in the next section.

6.3 Procedure of Application

This section describes the procedure used to implement the methodology described above within the ArcGIS 9 geoprocessing environment. This procedure includes a data model to support schematic navigation and attribute storage, an algorithm for sorting the schematic features from upstream to downstream, and the actual implementation of the procedure with an ArcGIS geoprocessing tool called ProcessSchematic.

6.3.1 Data Model

The ProcessSchematic tool operates on two geodatabase object classes: a schematic node class and a schematic link class. These classes may be feature classes or object classes, as the geometry of the nodes and links is not explicitly taken into consideration by the ProcessSchematic tool. Rather, the tool uses attributes within the node and link classes to establish connectivity between them. Thus, any two feature classes or tables may be used with the ProcessSchematic tool, as long as they are located in the same geodatabase. Note that if the schematic nodes and links are stored in tables, those tables must be registered with the geodatabase.

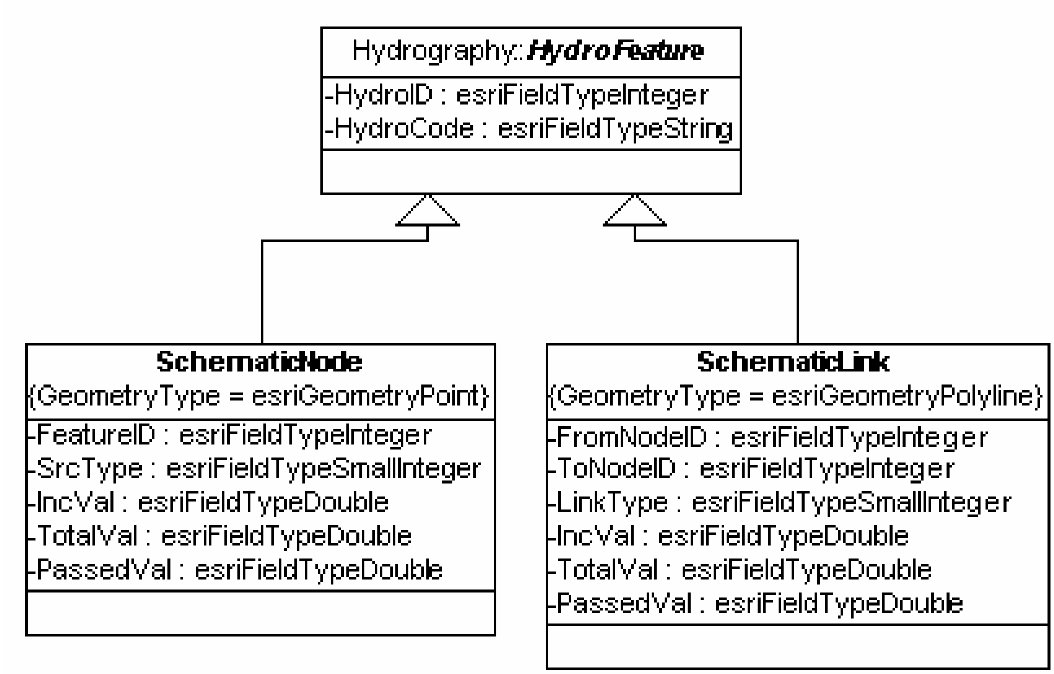


Figure 6.9 Data Model Required For Schematic Processor

The ProcessSchematic tool was designed to operate on an Arc Hydro schematic network, which uses feature classes to represent schematic nodes and links. Therefore, while the technically correct description of a schematic element in a standalone table is a *row*, the term *feature* will be used to describe schematic elements, as that term provides a better conceptual context of what a schematic element represents. In other words, schematic nodes and schematic links are referred to as schematic features, whether or not they belong to a feature class or a table.

Because the ProcessSchematic tool was designed to operate on an Arc Hydro schematic network, the tool assumes that the Arc Hydro schematic attributes exist in the schematic node and schematic link classes. These attributes are used for identifying schematic features and navigating the schematic network. For **Schematic Nodes**, these attributes include:

- HydroID - Long integer providing a unique identifier for the feature in the geodatabase
- HydroCode - Common text identifier for the feature
- FeatureID - Long integer identifying the hydro feature in the geodatabase to which the schematic feature corresponds
- SrcType - Short integer, user-defined, which indicates the type of the schematic feature, populated by Node/Link Schema Generation tool in the Arc Hydro tools

For **Schematic Links**, these attributes include:

- HydroID - Unique identifier for the feature in the geodatabase
- HydroCode - Common text identifier for the feature
- LinkType - Short integer, user-defined, which indicates the type of the schematic feature, populated by Node/Link Schema Generation tool in the Arc Hydro tools
- FromNodeID - The HydroID of the schematic node located at the upstream end of the schematic link
- ToNodeID - The HydroID of the schematic node located at the downstream end of the schematic link

In order for the ProcessSchematic tool to work correctly, all HydroIDs for the schematic features must be unique within the geodatabase.

In addition to the Arc Hydro attributes, schematic features may also possess attributes that describe the values that those features operate with. For both the schematic node and link classes, these optional attributes include:

- Incremental Value - Stores the incremental value for the given feature
- Total Value - Stores the total value for the given feature
- Passed Value - Stores the value that the feature will pass to the next downstream feature in the network

By default, the ProcessSchematic tool obtains values to be processed in the schematic network from the incremental value field. **Any values in the total value field or the passed value field will be overwritten by the new values calculated by the tool.** If the user does not wish to store the total value or the passed value in the attribute table for the schematic features, then these attributes need not be provided to the ProcessSchematic tool. The incremental value field is also left as an optional attribute, in case the user wishes to perform schematic processing that requires greater complexity than using the incremental value field as a starting point for obtaining values in the network.

Depending on what type of process is specified for each feature, other attributes may be required. For example, when decaying loads along a schematic link, travel time and decay constants may be required. In these cases, the attributes may be located on the features to which they pertain. These attributes are not read directly by the ProcessSchematic tool, but by the specific processing engine designed to work with those attributes. Processing engines are described in later in this chapter.

6.3.2 Processing Procedure

This section describes the procedure used by the ProcessSchematic tool for working with schematic networks. The procedure may be summarized into two major components: Data Preparation and the Process Loop.

In Data Preparation, the features in the schematic network are sorted from upstream to downstream, and collections are initialized that will store values during the processing of the network.

Once the features have been sorted, each feature is processed in the correct order until all features have been processed. Two types of processes may occur: a **receive process** and a **pass process**.

In a receive process, values received from upstream features are processed along with the incremental value for a given feature to produce a total value for that feature. By default, a receive process invokes a simple accumulation, so that all values received by a feature are added to its incremental value to produce the total value for that feature. If more complex processing is required (such as decaying loads), a DLL may be provided as a processing engine to perform the receive process.

In a pass process, the total value for a given feature is processed to produce a value that the feature will pass to the next downstream feature. By default, a pass process simply passes the total value for a given feature to the next downstream feature. If more complex processing is required, a DLL may be provided as a processing engine to perform the pass process.

Data Preparation

Before processing can begin, the schematic features must first be sorted so that features are processed from upstream to downstream. Consider the schematic network in Figure 6.10. In this network, Type 1 features are colored green, and Type 2 features are colored red. The number shown by each feature represents the HydroID for that feature. Schematic nodes have HydroIDs indexed at 1000, while schematic links are indexed at 4000. The features at the top of the diagram represent the most upstream features, while the feature at the bottom of the diagram (i.e. schematic node 1007) represents the most downstream feature.

Using the FromNodeID and ToNodeID attributes on SchematicLink, a Sort Collection is built containing the schematic features in order from upstream to downstream, where a collection is a Visual Basic object containing a set of values or other objects. The order within the Sort Collection is not necessarily unique. For example, for the schematic network shown in Figure 6.10, the Sort Collection's first entry might be the feature with HydroID = 1001, followed by the feature with HydroID = 4002, and so on; or the collection could start with HydroID = 1002, then 4001, and so on. The algorithm for building the Sort Collection chooses its starting branch at random, and merely insures that all upstream branches are accounted for before adding a downstream node or link to the collection.

A Topology Collection is also created at this time. For each feature, this collection stores the HydroIDs of adjacent upstream features, indexed by the HydroID of the current feature. For example, the feature with HydroID = 1003 will have associated HydroIDs of 4002 and 4004 in the Topology Collection.

Finally, a Value Collection is created to store the values that each feature will pass to the next downstream feature. For example, the schematic node with HydroID = 1001 might have a value of 30 in the Value Collection, which may represent the flow that the node sends downstream.

These collections serve as efficient tools in processing the schematic network.

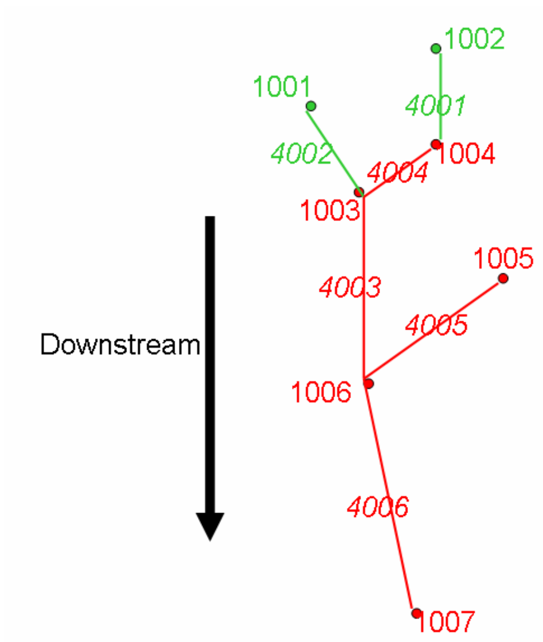


Figure 6.10 Sample Schematic Network

The Process Loop

Once the features in the schematic network have been sorted, each feature is processed in the correct order. For each feature, upstream values are processed and combined with the incremental value for that feature using a receive process. The combined value is then processed using a pass process to produce a value that will be passed to the next downstream feature, and so on. There are five steps in the process loop:

1. Get upstream features
2. Get upstream values
3. Process upstream and incremental values
4. Process current value to pass downstream
5. Update Value Collection with value to pass

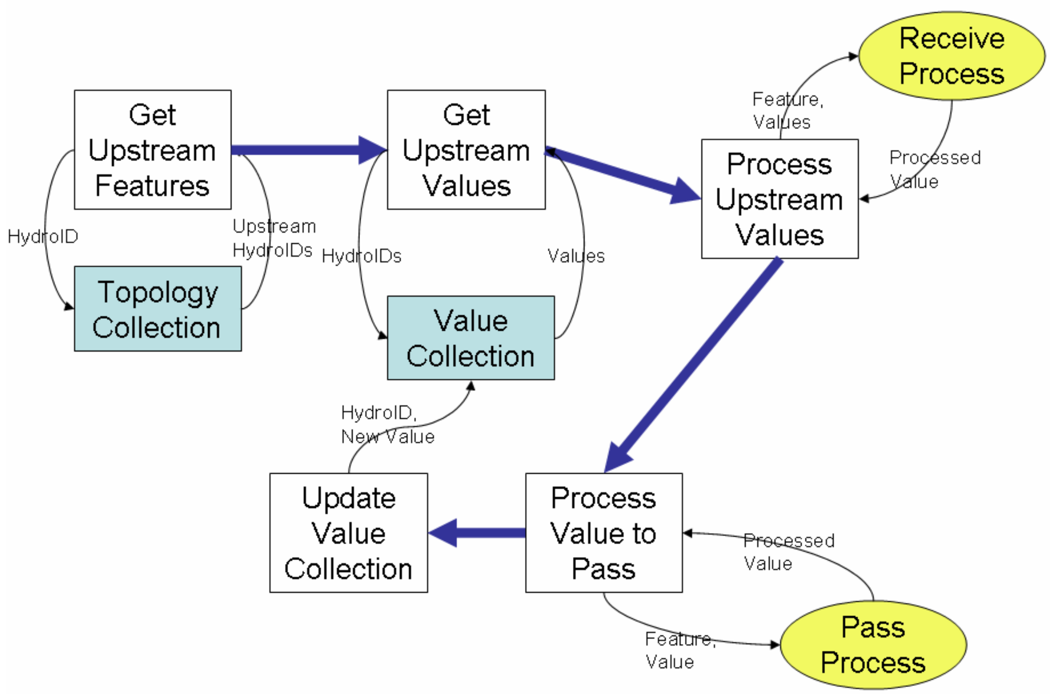


Figure 6.11 Summary of Single Iteration in the Process Loop

To illustrate the process loop, an example using the schematic link with HydroID = 4003 from the network in Figure 6.10 is given.

1. Get Upstream Features

From the topology collection, we find that schematic node 1003 is upstream of schematic link 4003.

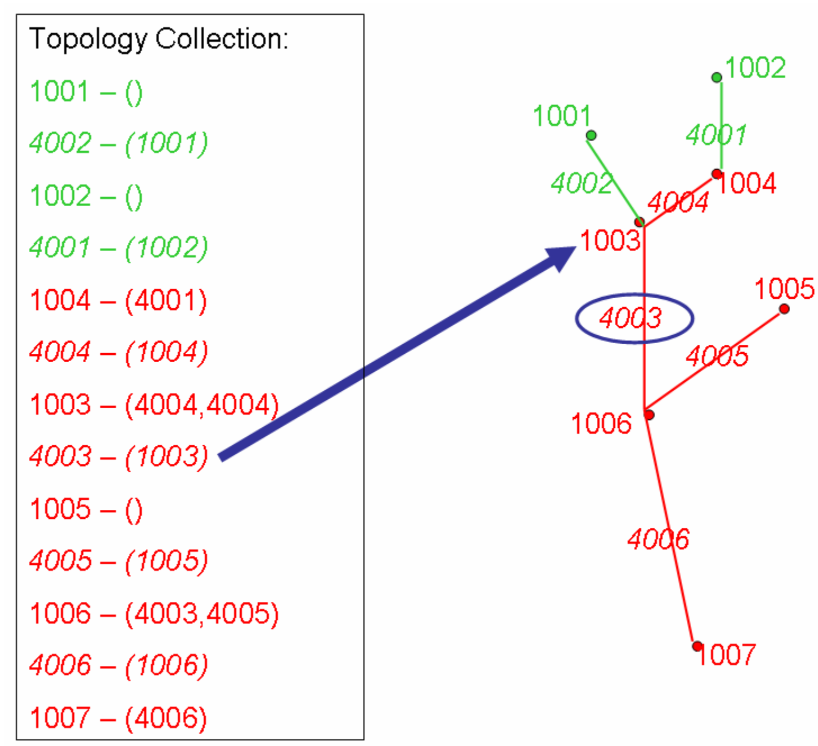


Figure 6.12 Accessing the Topology Collection

2. Get Upstream Values

From the Value Collection, we find that schematic node 1003 has a value to pass of 3.

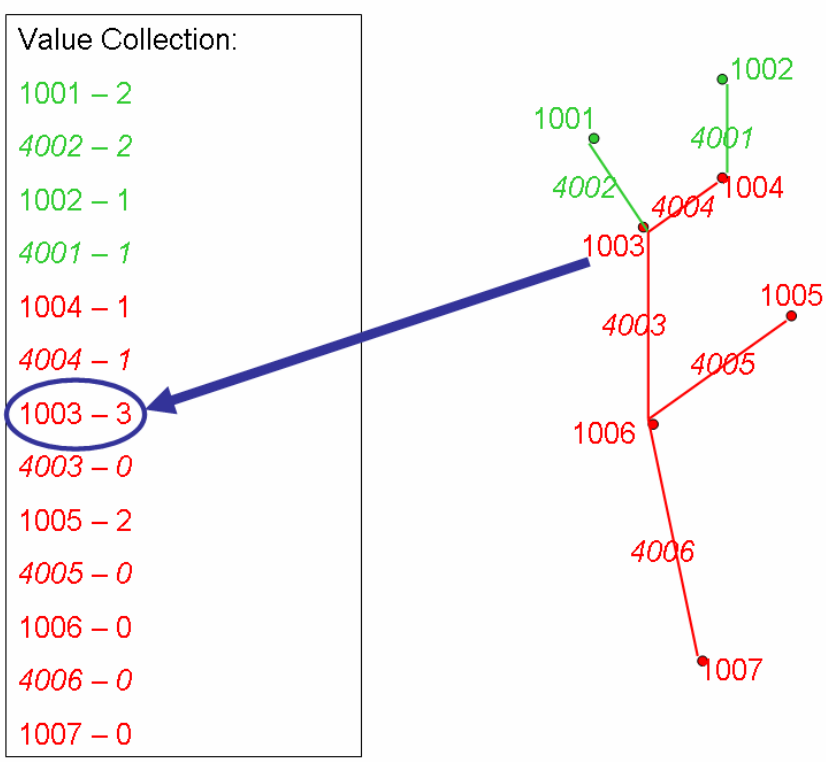


Figure 6.13 Accessing the Value Collection

3. Process Upstream and Incremental Values

The appropriate processor for this feature is identified by noting that this is a Type 2 schematic link, and that the process to be called is a receive process. Once the processor is identified, it is called to perform the receive process. The resulting value from the receive process is stored in the attribute table for the current feature, in the total value field (if supplied).

4. Process Value to Pass Downstream

The appropriate pass processor is called for this feature, given that this is a Type 2 schematic link.

5. Update Value Collection

The Value Collection for the current feature is updated, so that the entry associated with this feature contains the value that this feature will pass to the next downstream feature. The resulting value from the pass process is stored in the attribute table for the current feature, in the passed value field (if supplied).

6.3.3 Implementation with ArcToolbox

The implementation of this procedure in ArcToolbox requires the following components:

- ProcessSchematic - **Script tool** within an ArcGIS 9 toolbox
- ProcSchematic.vbs - **Script** (written in VBScript) on disk, referenced by ProcessSchematic script tool
- MBSchematic.dll - **DLL** (written in VB) that sorts the schematic features and handles the calling of process DLLs
- Process DLLs - zero or more **DLLs** that may be called to process schematic features

Both scripts and DLLs are used in this research. The majority of processing in the methodology described above occurs within a DLL. The script is required to provide a bridge between ArcToolbox and the DLL, as ArcToolbox script tools may only be linked to a script, and not a DLL. Thus, the interface provided by the ArcToolbox script tool manages the input and output data, the DLL performs the processing work, and the script serves as the go-between.

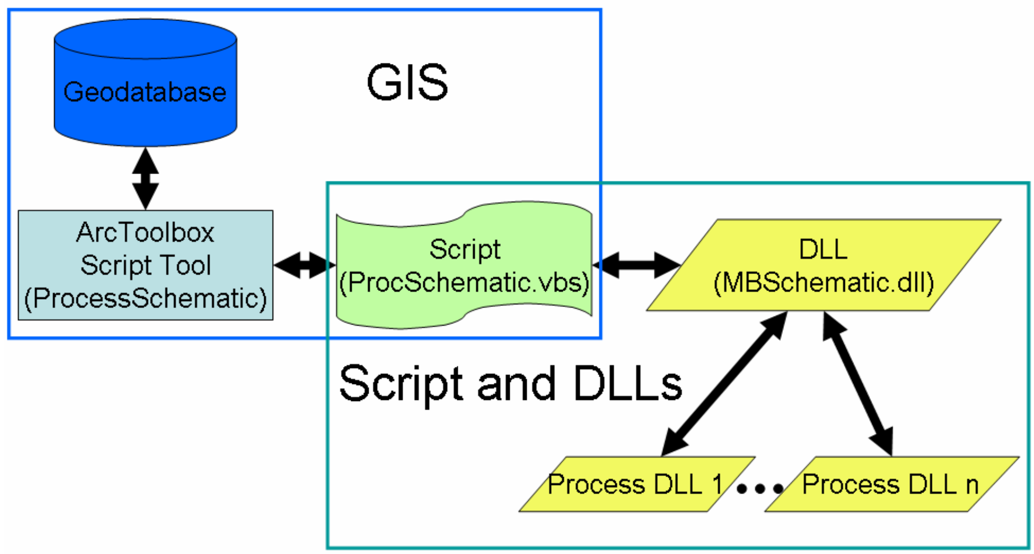


Figure 6.14 Implementation of Methodology Using Scripts and DLLs

ProcessSchematic Script Tool

The ProcessSchematic script tool was created in a custom toolbox for ArcGIS 9.0. This tool is linked to the script ProcSchematic.vbs, and passes the following parameters to that script:

- Schematic link feature class
- Incremental value field in schematic link feature class (read only)
- Total value field in schematic link feature class (write only)
- Passed value field in the schematic link feature class (write only)
- Schematic node feature class
- Incremental value field in schematic node feature class (read only)
- Total value field in schematic node feature class (write only)

- Passed value field in the schematic node feature class (write only)
- List of processing engines
- List of source types for the processing engines
- List of feature types for the processing engines
- List of behavior types for the processing engines

Recall that the tool reads information from the incremental value fields, and if specified, writes information to the total and passed value fields. Thus, the values in the incremental value fields are not altered by the tool. Also, the tool reads no information from the total or value fields. It simply overwrites any existing information in those fields with new information calculated while processing the schematic network.

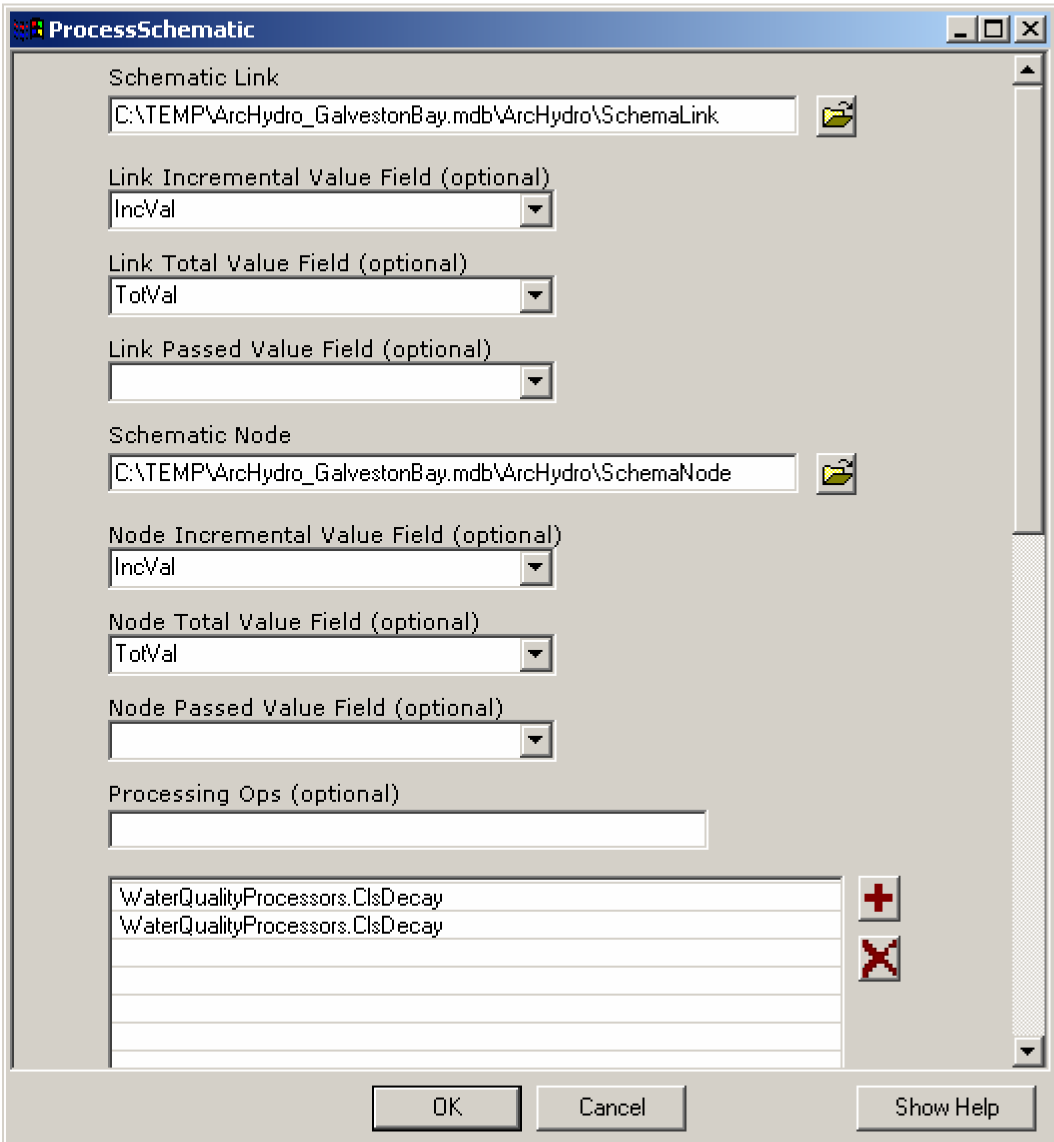


Figure 6.15 Feature and Processing Engine Inputs for ProcessSchematic Script Tool

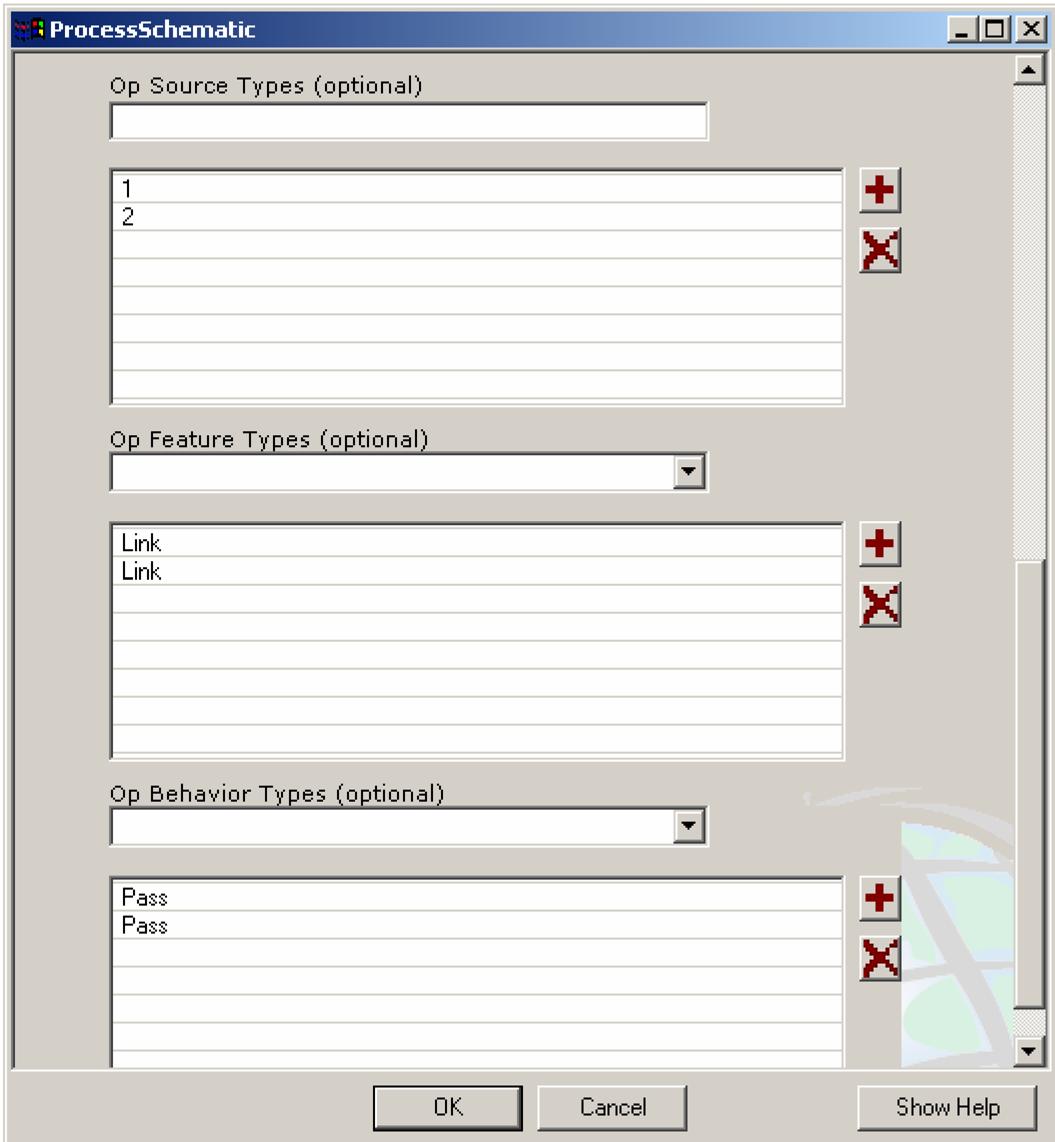


Figure 6.16 Processing Engine Descriptors for ProcessSchematic Script Tool

The classes in the DLLs that will perform advanced processing are specified in the Processing engines list. Each processing engine must be associated with a specific source type, feature type, and behavior in the script tool.

The first entry in the list of processing engines corresponds to the first entry in the list of source types, the first entry in the list of feature types, and the first entry in the list of behavior types. The second entry in the list of processing engines corresponds to the second entry in the list of source types, and so on. Thus, the number of entries in the lists of source types, feature types, and behavior types must match the number of entries in the list of processing engines.

The feature type defines whether the processing engine is associated with schematic links or schematic nodes. The source type defines the type of schematic node (indicated by SrcType field) or schematic link (indicated by LinkType field) that the processing engine is associated with. Finally, behavior type defines whether the processing engine is associated with receiving behavior or passing behavior. Thus, each processing engine is only used in a very specific instance. For example, processing engine that decays loads might only be used with schematic links of Type 2 when values are being passed from a schematic link to the next downstream feature.

If no processing engines are given for a combination of feature type, source type, and behavior (passing/receiving), then the default behavior is a simple accumulation for receiving operations, and passing the total value (without changing the value) for passing operations.

The script tool also receives two parameters from the script:

- Success: Boolean - True if operation was successful
- Abort: Boolean - True if operation failed or was canceled

The two parameters allow the script tool to be chained together in a sequence with other tools in an ArcToolbox model. They do not affect the execution of the script.

ProcSchematic Script

This VBScript script takes the inputs from the ProcessSchematic script tool and converts them into a format that the MBSchematic DLL can recognize. The script also creates an instance of clsProcessSchematic, a class from the MBSchematic DLL that performs most of the work in processing the schematic network. The script calls the ProcessSchematic function from clsProcessSchematic, and feeds the class the parameters from the script tool. When the class is finished, the script determines if the operation was successful, and sends the appropriate information (Success or Abort) back to the script tool.

MBSchematic DLL

This DLL performs most of the work outlined in this methodology, including data preparation and control of the process loop. The DLL contains a class called clsProcessSchematic with a function called ProcessSchematic that may be called from a script. The function accepts inputs about the schematic network and the processing engines. The MBSchematic DLL controls the instantiation of each of the processing engines specified in the script tool, and determines when each processing engine is called based on the other script tool parameters. After sorting the features, the DLL loops through each feature. If a feature meets the criteria for calling a specific op, the DLL passes that feature and the appropriate values to that processing engine for processing.

The MBSchematic DLL also handles the writing of data (i.e., the values in the Total Value field) to the geodatabase. However, because each processing engine receives an IRow object as one of the parameters, the processing engines also have the ability to write values to the geodatabase, if such behavior is necessary.

Processing Engines

Zero or more processing engines may be used to process the schematic network. The ops are specified in the script tool, along with the conditions for their use (i.e., source type, feature type, and behavior type.) Each processing engine is specified by giving the DLL name, followed by a period, and then followed by the name of the class in the DLL that will perform the operation. For example, a DLL called WaterQualityProcessors with a class called ClsDecay would be specified as WaterQualityProcessors.ClsDecay. Note that this DLL must be registered on the computer in order for the script to locate it.

Each class in a DLL that will serve as a processing engine must contain a public function with the following signature:

```
Public Function ProcessVals(pRow As IRow, colVals As
Collection) As Double
```

The function must be called "ProcessVals", and it must have the two arguments of type IRow and Collection. The first argument represents the schematic feature that is to be processed. The second argument is a Visual Basic Collection object that stores the values that are to be processed for this schematic feature. Each object in the Collection is an array representing a value from an upstream feature and information needed to obtain that upstream feature. The first value in the array (at the lowest index) is the value from the upstream feature. The second value is the name of the feature class to which that upstream feature belongs. The third value is the ObjectID of that upstream feature. The function must return a value of type Double.

Following this structure gives the user the ability to create as many processing engines as desired. The MBSchematic DLL, which handles instantiating and calling the ops are the correct time, does not know or care how the processing engines do their work. MBSchematic simply passes the ops some information, and expects some information in return. The processing engine could be as simple or as complex as needed.

6.4 Case Studies

Two case studies are presented, which use the schematic processor to model hydrologic behavior with features in a GIS. The first simulates rainfall-runoff response and channel routing using routines from HEC's library of hydrologic functions, LibHydro. The second simulates bacterial loading in Galveston Bay from point and nonpoint sources.

6.4.1 Schematic Network Case Study I: LibHydro Application

The HEC has produced hydrologic simulation models that have been utilized in many applications throughout the years. The HEC recognized that certain functions in those models could be extracted to provide useful routines applicable in many situations outside of the HEC modeling environment. This led to the development of a library of functions called LibHydro. These functions were derived from the HEC-1 software, which is the predecessor to HMS. The routines are coded in FORTRAN, and can be accessed by calling the LibHydro DLL using a variety of programming languages, including Visual Basic and C++. Examples of routines include unit conversions and Muskingum routing (HEC, 1995). The number of routines in LibHydro is still growing as the HEC identifies and incorporates more functions into the DLL. By referencing LibHydro, a hydrologic application can make use of effective modeling routines that have already been proven through years of use.

Masatsugu Takamatsu and Tim Whiteaker at the University of Texas at Austin developed a schematic network application, which uses functions from LibHydro to simulate rainfall-runoff and routing calculations for a portion of the Llano River basin in Texas. Takamatsu was the primary developer of the application, while Whiteaker provided instruction on how to use the schematic processor, programming assistance, and overall guidance.

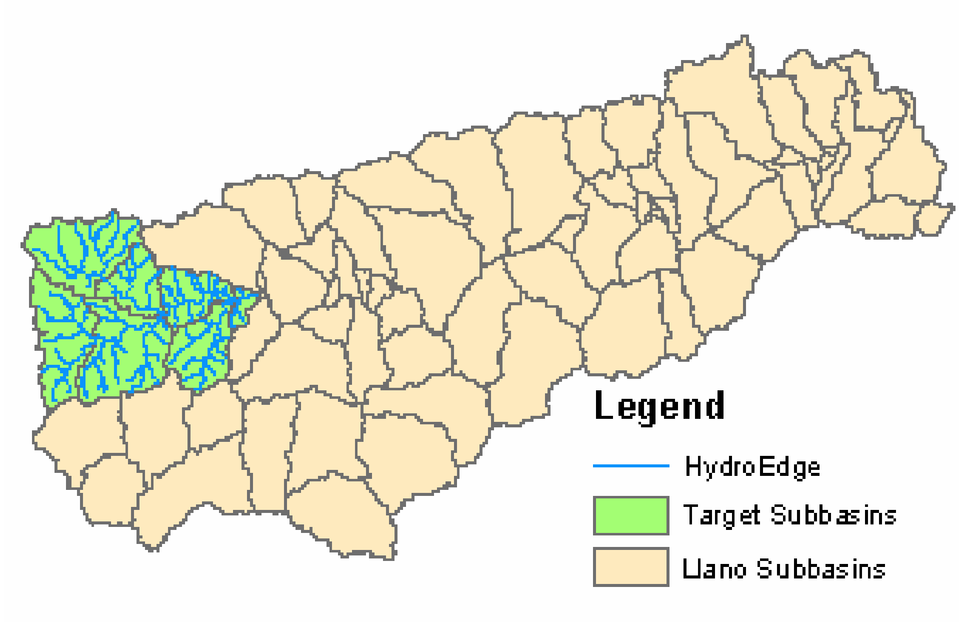


Figure 6.17 Llano Basin and Target Subbasins

The raw data for this application included a Watershed feature class representing the target subbasins for the analysis, precipitation data stored in the Arc Hydro TimeSeries table, and a HydroNetwork composed of HydroEdges and HydroJunctions. The precipitation data covered a period from November 2nd through November 8th, 2000, at 15-minute intervals.

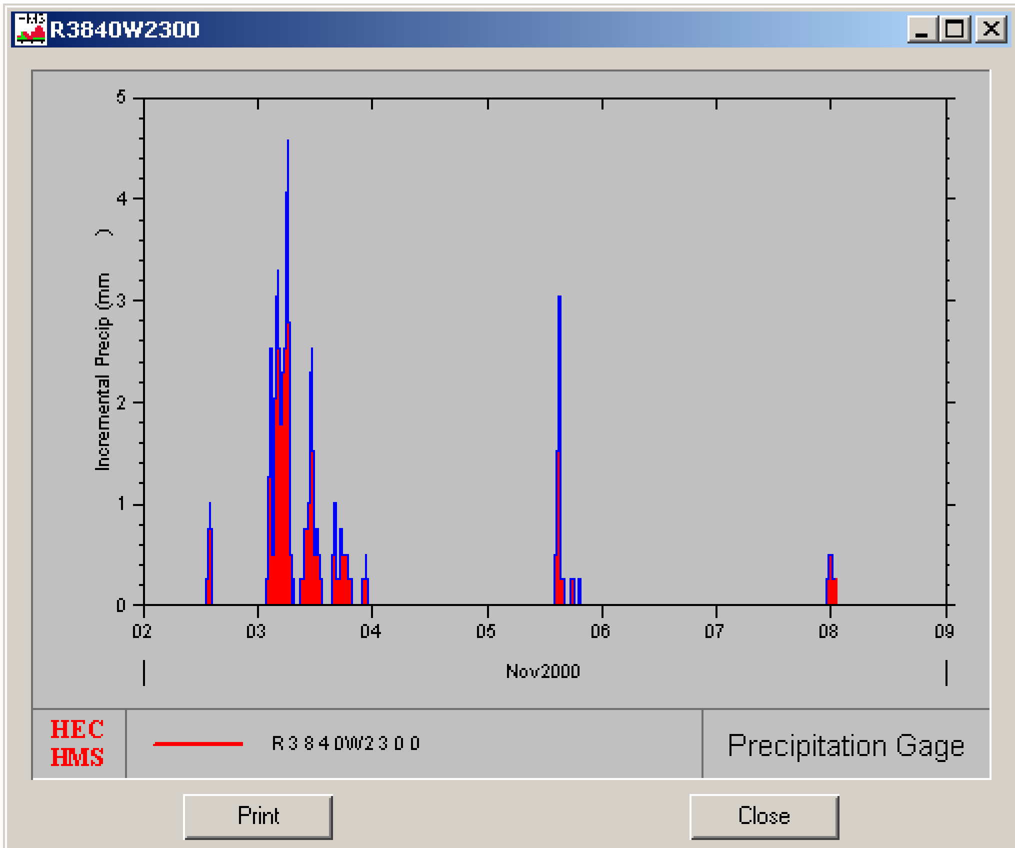


Figure 6.18 HMS Plot of Precipitation Data in the Llano Basin

A schematic network was built using the Arc Hydro tools. The schematic network includes nodes representing Watersheds (Type 1 nodes) and HydroJunctions (Type 2 nodes), and links connecting those nodes. Type 1 links connect Type 1 nodes to Type 2 nodes, while Type 2 links connect Type 2 nodes to Type 2 nodes.

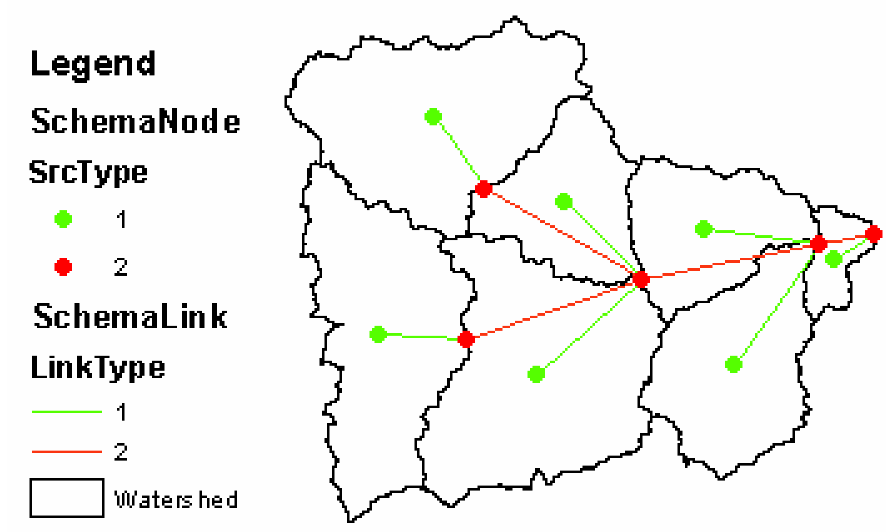


Figure 6.19 Schematic Network for Llano Target Subbasins

Four LibHydro functions were used in this analysis: Initial Constant Loss, Snyder Unit Hydrograph, Baseflow, and Modified Puls Routing. The first three functions were used to prepare time series of watershed outflows to the stream network, while the last function was used by the schematic network to route the flows to the outlet of the subbasin network. These functions were incorporated into an ArcGIS 9 model called Rainfall to Routed Flow. More information about the specifics of each LibHydro function can be found from Takamatsu (2003).

Function	Purpose
Initial Constant Loss	Calculate precipitation excess Using initial loss and constant loss rate
Snyder Unit Hydrograph	Calculate Watershed outflow from precipitation data using Snyder unit hydrograph
Baseflow	Calculate baseflow using HEC-1 baseflow method
Modified Puls Routing	Route channel flow using Modified Puls routing method

Table 6.2 LibHydro Functions Used by 'Rainfall to Routed Flow' Model

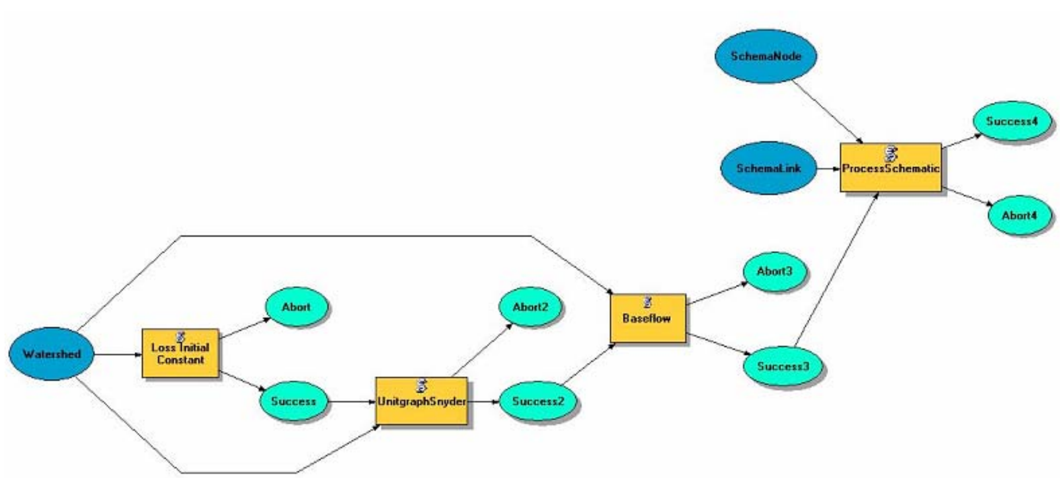


Figure 6.20 Full View of Rainfall to Routed Flow Model

The Rainfall to Routed Flow model consists of four script tools, one associated with each of the LibHydro functions listed above. The first three tools prepare time series information to be used by the Schematic Processor for routing the flows. The last tool is the Schematic Processor. Each tool accesses a DLL called MBlibHydro in order to use the functions in LibHydro. These tools are described below.

Loss Initial Constant

The Loss Initial Constant tool calls the Loss Initial Constant function from LibHydro to calculate precipitation excess for each Watershed using rainfall data from the TimeSeries table. Fields in the Watershed feature class provide the values of initial loss, constant loss rate, and impervious area ratio for each Watershed. The tool returns a value of Success=True if the operation was successful.

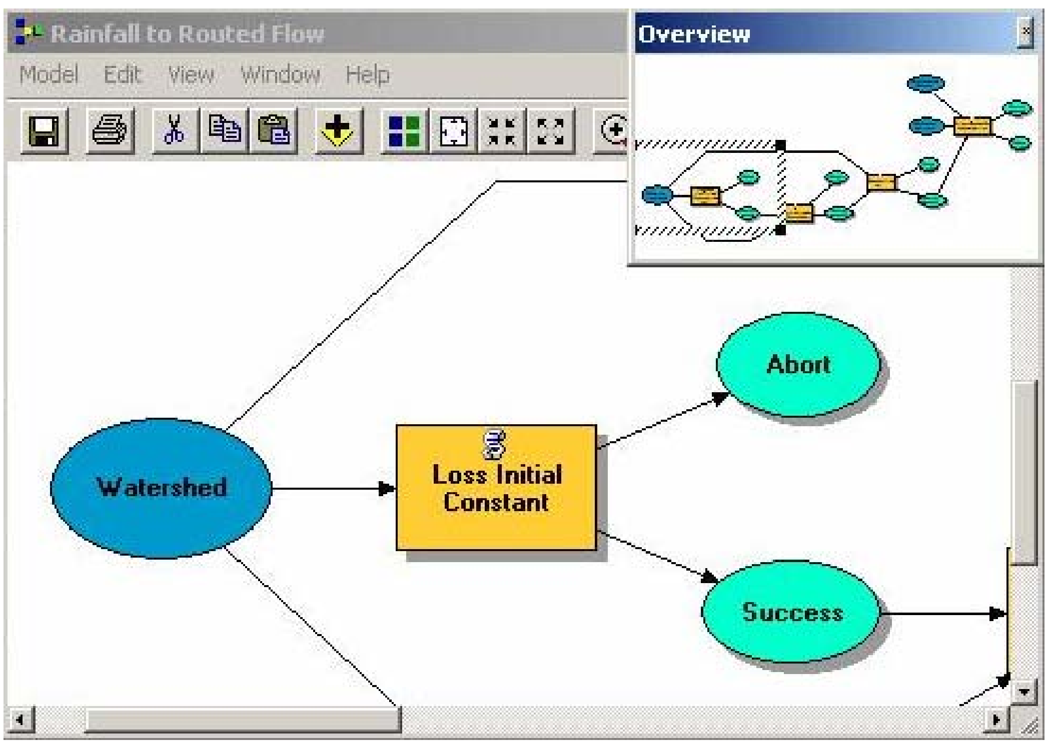


Figure 6.21 Loss Initial Constant Tool

UnitgraphSnyder

The UnitgraphSnyder tool calls the Snyder Unit Hydrograph function from LibHydro to calculate a runoff hydrograph for each watershed, given the precipitation excess time series determined by the Loss Initial Constant Tool. Fields in the Watershed feature class provide the values of Snyder Cp, Snyder Tp, and Basin Area for each Watershed. The UnitgraphSnyder tool only executes if the Loss Initial Constant tool completed its operations successfully. UnitgraphSnyder returns a value of Success=True if the operation was successful.

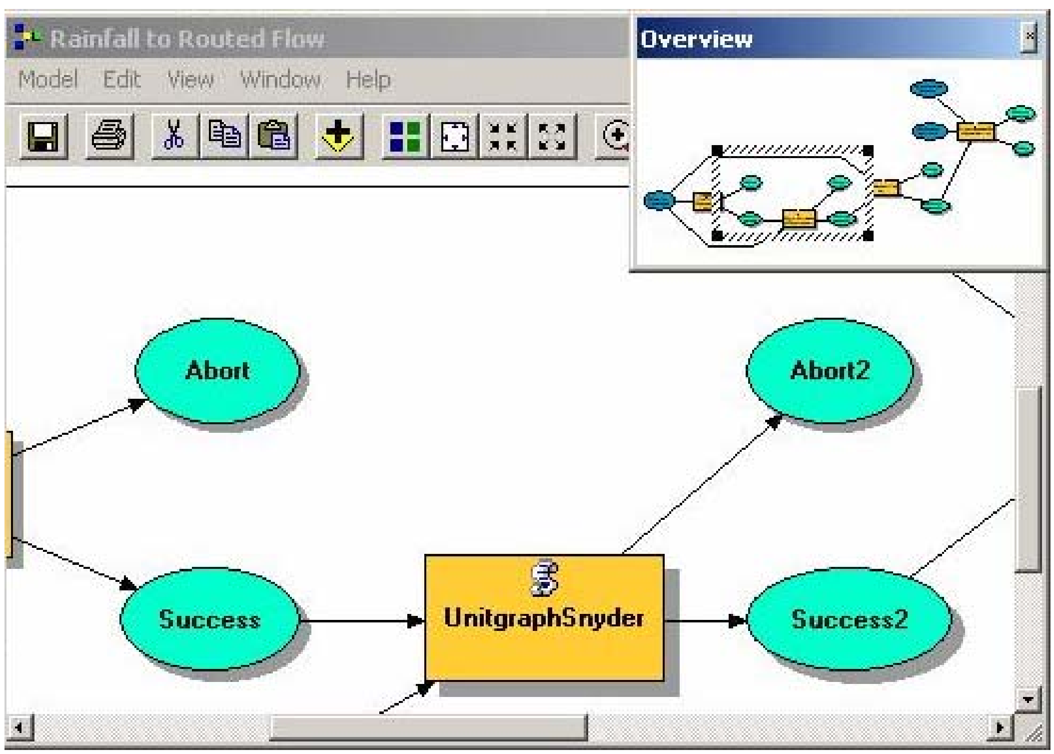


Figure 6.22 UnitgraphSnyder Tool

Baseflow

The Baseflow tool calls the Baseflow function from LibHydro to calculate Baseflow for each watershed, given the precipitation excess time series determined by the Loss Initial Constant Tool. The tool then adds the baseflow to the runoff hydrograph to produce an outflow time series for each watershed. Fields in the Watershed feature class provide the values of recession ratio and recession threshold for each Watershed. The Baseflow tool only executes if the UnitgraphSnyder tool completed its operations successfully. Baseflow returns a value of Success=True if the operation was successful.

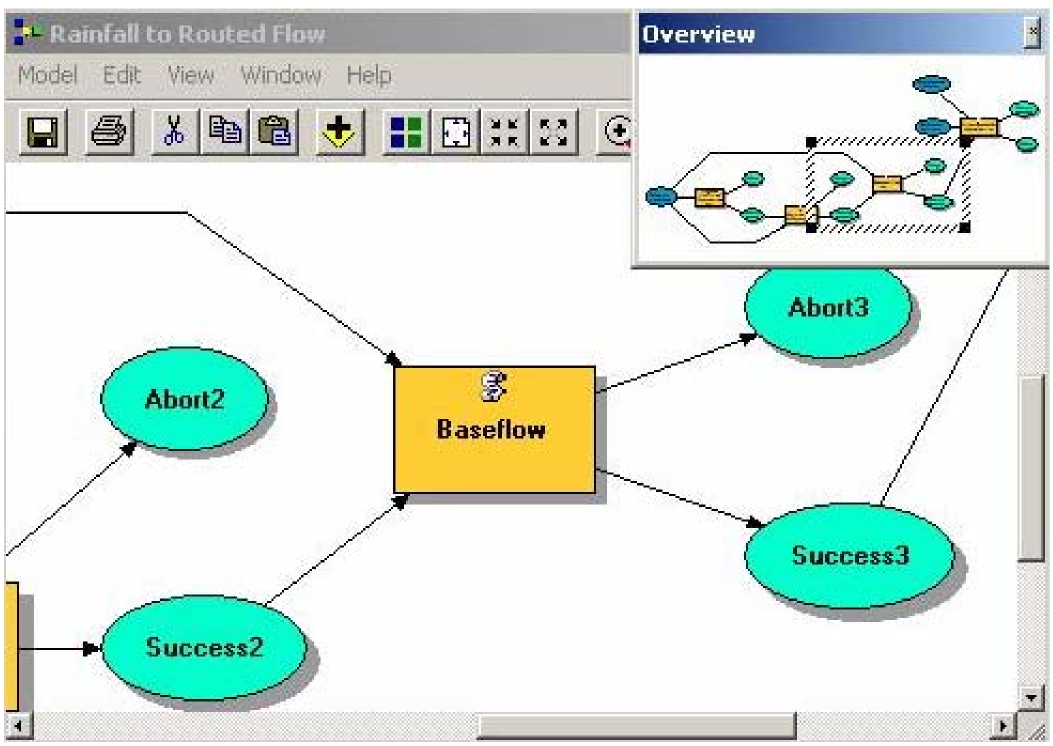


Figure 6.23 Baseflow Tool

ProcessSchematic

Once outflow time series have been calculated for each watershed, the ProcessSchematic tool uses the schematic network to route the flows through the stream network to the outlet of the target subbasins. This particular application of the Schematic Processor uses the tool in a unique way. The tool is designed to pass single values from feature to feature. However, the Rainfall to Routed Flow model requires time series of flows to be passed between features, rather than a single value. Therefore, instead of passing single values between features, the Schematic Processor is set up to pass time series type IDs, or TSTypeIDs, which categorize values in the Arc Hydro TimeSeries table (such as rainfall, runoff, etc.). SchemaLinkPuls.DLL and SchematicNode.DLL are set as processing engines, whose purpose is to extract time series data for each feature based on the TSTypeID, route the time series, and write the resulting time series of routed flows back to the TimeSeries table. SchemaLinkPuls.DLL uses the Modified Puls Routing function from LibHydro to route flows along a river reach, represented by a schematic link of Type 2. When two reaches converge, their time series of routed flows are added together to produce a combined flow. SchematicNode.DLL performs that task, with the assumption that there are no backwater effects.

The ProcessSchematic tool only executes if the Baseflow tool completed its operations successfully. ProcessSchematic returns a value of Success=True if the operation was successful.

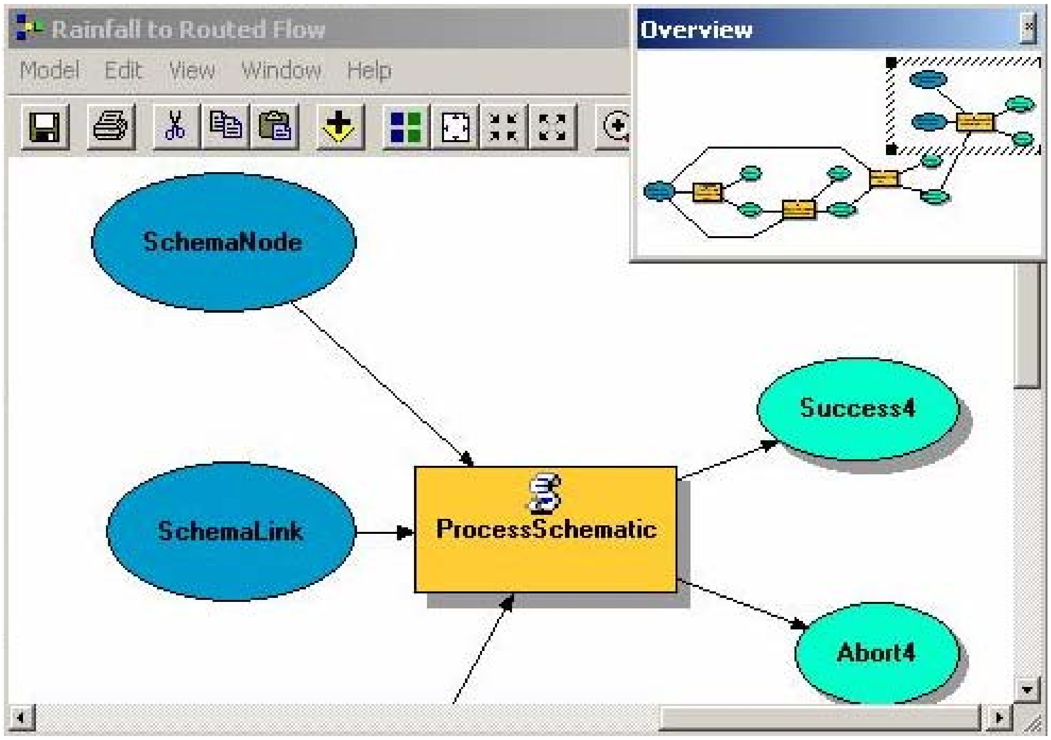


Figure 6.24 ProcessSchematic Tool

Once the Rainfall to Routed Flow model has completed execution, a time series of outflow values for the basin comprised of the target Llano subbasins is recorded in the Arc Hydro TimeSeries table.

The Rainfall to Routed Flow model was run for the Llano subbasins covering precipitation events from November 2nd through the 8th in the year 2000. The same model was setup and run within HEC-HMS.

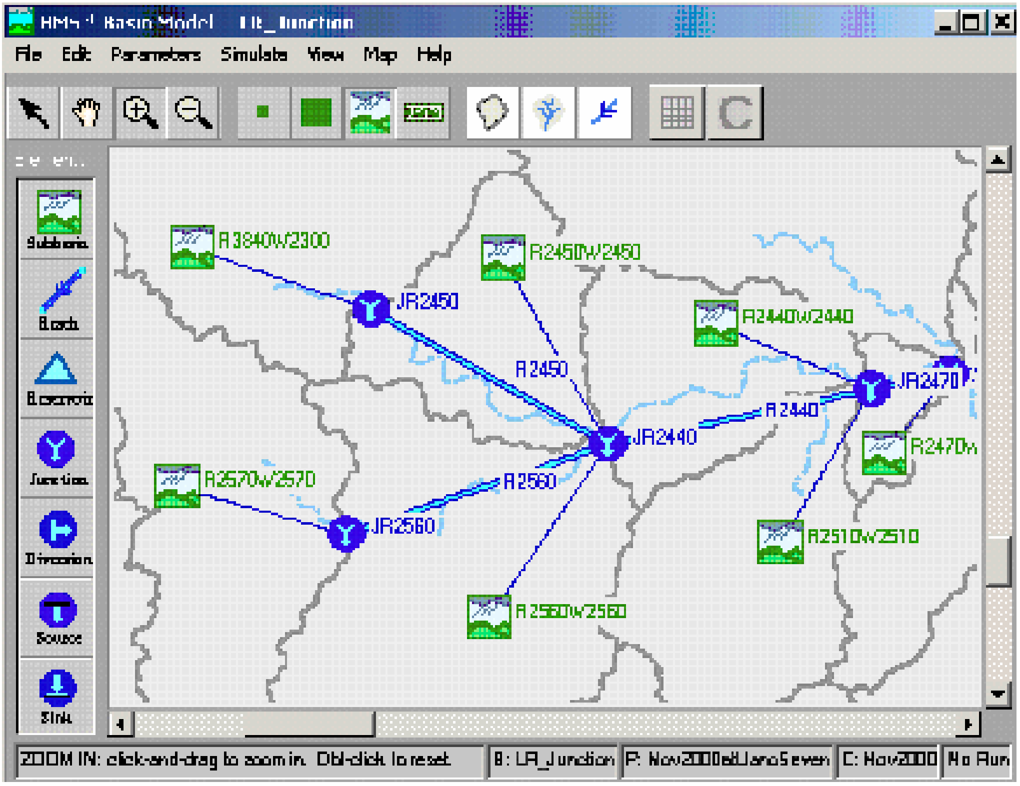


Figure 6.25 HMS Basin Model for Llano Target Subbasins

The results from both models matched almost exactly, with differences of less than one-half of one percent in most cases. Takamatsu and Whiteaker suggest that rounding within each model may account for the differences between the two models. Still, with nearly identical results, this experiment verifies that the Rainfall to Routed Flow model is successfully in calling the same functions used by HMS.

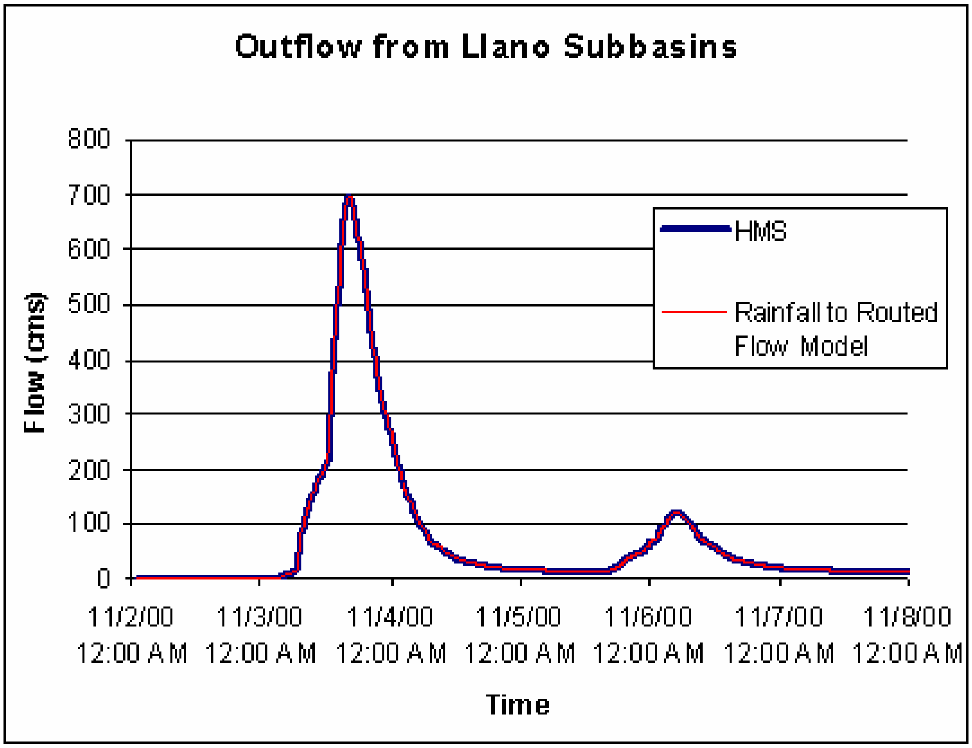


Figure 6.26 Outflow Hydrographs Computed by HMS and Rainfall to Routed Flow Model

Timestamp	Flow (cms) from HMS	Flow (cms) from Rainfall to Routed Flow Model	Percent Difference (%)
11/3/00 6:45 PM	577.980	577.785	0.034
11/3/00 7:00 PM	560.730	560.627	0.018
11/3/00 7:15 PM	542.840	542.861	0.004
11/3/00 7:30 PM	523.750	523.928	0.034

Table 6.3 Values Computed by HMS and Rainfall to Routed Flow Model Varied by Fractions of a Percent

Another important observation from the execution of these two models is that HMS performed the simulation much quicker than the Rainfall to Routed Flow model. A complete run of Rainfall to Routed Flow took approximately 20 minutes to finish. The HMS model took only 20 seconds to run.

The Rainfall to Routed Flow model proves that hydrologic and hydraulic calculations comparable to that possible with simulation models such as HMS can be carried out within a GIS application by accessing functions from LibHydro. The work could be extended to include functions from other object libraries. The modular nature of ModelBuilder tools makes the components in Rainfall to Routed Flow extensible and reusable.

One important find from this experiment was that the creation and manipulation of large amounts of time series data is very costly in time when writing to a geodatabase. For these types of applications, the best approach may be to write only the final results of model simulations back to the Arc Hydro TimeSeries table. The intermediate time series can be calculated through LibHydro and managed in a more efficient medium, or HMS itself can be called from ModelBuilder to perform the simulation, returning control to the GIS when the simulation is completed.

6.4.2 Schematic Network Case Study II: Water Quality Modeling in Galveston Bay

Whiteaker and Goodall (2003) applied the schematic processor to water quality modeling in Galveston Bay, Texas, by examining fecal coliform loading from point and nonpoint sources into the bay. The procedure and equations used to model bacterial loadings were adapted from Zoun (2003), who used the Arc Hydro tools and conventional GIS techniques to determine loadings. Goodall developed three processing engines for the schematic processor to simulate bacterial load transport, while Whiteaker assembled an ArcGIS workflow model, WQModel, to perform the analysis. WQModel includes components to determine watershed loads, as well as the schematic processor to accumulate loads into the bay. WQModel requires general data development, such as watershed delineation and the calculation of travel times on reaches, to have already been accomplished and verified. The schematic network must also have already been created with three types of nodes and links:

Nodes

- Type 1: Represents watersheds
- Type 2: Represents junctions in the stream network
- Type 3: Represents bays

Links

- Type 1: Links watersheds to stream network (Type 1 nodes to Type 2 nodes)
- Type 2: Links river segments (Type 2 nodes to Type 2 nodes)
- Type 3: Links river outlets to bay (Type 2 nodes to Type 3 nodes)

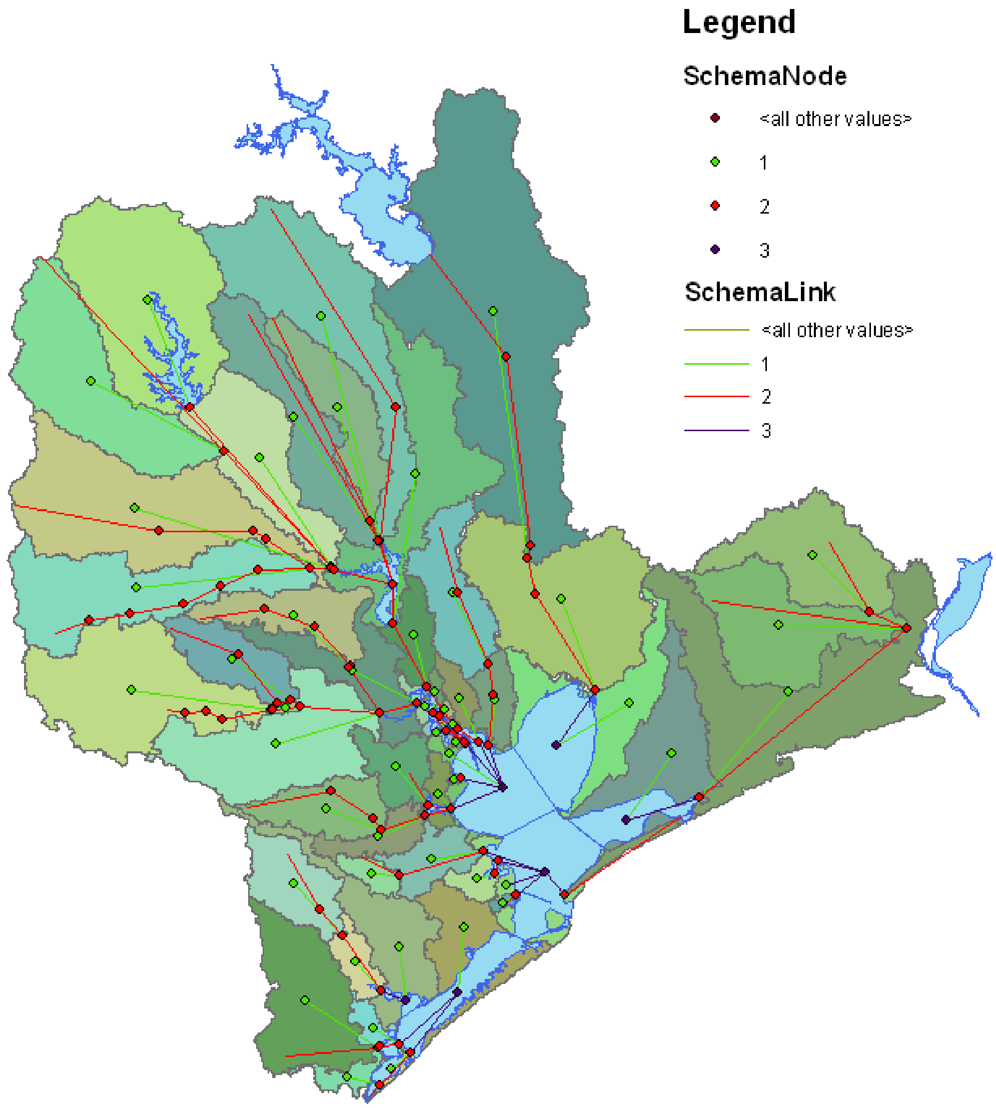


Figure 6.27 Gavleston Bay Watershed with Schematic Network (Whiteaker and Goodall, 2003)

The model then uses rainfall data and watershed characteristics such as land use to determine annual watershed loads. The schematic processor is then called to move the loads from the watersheds into the bay, decaying the loads as they travel through the stream network. The model contains fifteen tools, including standards ArcGIS tools as well as custom script tools that access DLLs. The reader is referred to Whiteaker and Goodall (2003) for a detailed explanation of each tool in the model.

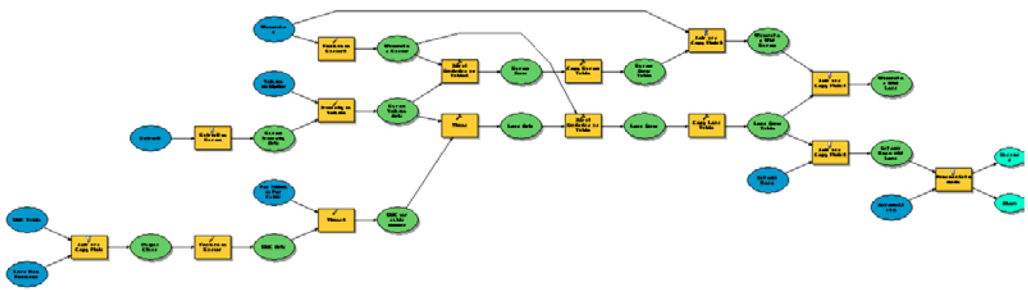


Figure 6.28 ModelBuilder diagram of WQModel

The three types of links and three types of nodes signify the behavior that features in the schematic network will implement. When behavior more complex than simple accumulation is required, the schematic processor calls functions from a WaterQualityProcessors DLL that was written to support water quality modeling. The processes associated with each type of link and node are shown in the tables below.

Links

Type	Purpose	Process	dll
1	land to river mass transport	Mass is passed to downstream node without decay because incremental value represents mass of bacteria which reaches watershed outlet.	none
2	along river transport	Mass passed to downstream node is decayed according to time of travel and decay coefficient.	clsDecay
3	river to bay	Mass is passed from river outlet to bay without decay.	none

Table 6.4 Explanation of processing engines for schema links (Whiteaker and Goodall, 2003)

Type	Purpose	Process	dll
1	losses or additions of mass for watersheds	Mass is passed to downstream link without decay.	none
2	losses or additions of mass along river network	Mass is passed to downstream link without decay except for one feature for which the LossCoef field is nonzero. The 0.5 factor is to account for bacteria that enters Sabine Lake from the upstream Watersheds for East Bay.	clsLossCoef
3	losses or additions of mass for bays	Mass is received by the bay and used to estimate concentration (cfu/m3) within the bay. Estimated concentration in bay is based on continuous flow stired tank reactor (CFSTR) assumptions.	clsCFSTR

Table 6.5 Explanation of processing engines for schema nodes (Whiteaker and Goodall, 2003)

The objects in the 'dll' column in the above tables are classes that exist in the WaterQualityProcessors DLL. These classes were constructed to function as schematic processors, and thus conform to the rules specified above for creating a processing engine. The classes are associated with the appropriate link or node types when the user specifies inputs to the schematic processor.

Three processing engines were constructed for this case study. The first op, clsDecay, used a first order decay rate and received loads to simulate the decay of bacteria along stream segments. The decay rate represents the loss of mass due to biological decay, sorption, uptake, etc, as material moves downstream. Decay rates for each stream segment were determined by Zoun (2003) and stored as attributes on the SchemaLink feature representing that segment. The *clsDecay* processing engine is called when a SchemaLink is passing a load to its next downstream node. The load passed is reduced according to a first-order reaction shown in Equation 6.1.

$$load_{passed} = load_{receieved} \times e^{-kt} \qquad \text{(Eq. 6.1)}$$

Where:

$load_{passed}$ = downstream bacteria load (cfu/yr)

$load_{received}$ = upstream bacteria load (cfu/yr)

k = first-order decay coefficient (day^{-1})

t = travel time (day)

The second processing engine, clsCFSTR, calculates a bay's increase in concentration associated due to an import of bacteria mass. Each bay's volume, estimated yearly flow, and first-order decay term are stored in the SchemaNode feature class. The load received by the SchemaNode representing the bay is used to estimate the bay concentration, assuming the bay has constant inflow equal to its outflow, and that mass entering the bay is instantaneous and perfectly mixed within the bay (Eq. 6.2). These are commonly known as continuous flow, stirred tank rector (CFSTR or CSTR) assumptions.

$$c = \frac{L}{Q + kV} \qquad \text{(Eq. 6.2)}$$

Where:

c = concentration within bay (cfu/m^3)

L = bacteria load entering bay (cfu/yr)

Q = total flow (m^3/yr)

k = first-order decay term (day^{-1})

V = volume of bay (m^3)

The third processing engine, clsLossCoef, accounts for bacteria from the watersheds upstream of East Bay that flows into Sabine Lake and not East Bay. From Zoun (2003), half of the load from these watersheds is assumed to flow into East Bay, while the other half is delivered to Sabine Lake. The clsLossCoef processing engine uses the LossCoef attribute of a SchemaNode feature to determine the fraction of the load from that node that is passed to its downstream link. This allows one to easily adjust the proportioning of load between East Bay and Sabine Lake. The LossCoef value for the appropriate nodes in the Galveston Bay database were assigned a value of 0.5.

Once the data were prepared and the processing engines defined, the WQModel was run. The results of WQModel execution match up quite well with Zoun's results. However, some discrepancies as high as 2.6% were present. These discrepancies were due to differences in the runoff grid used by Zoun and the one used during ModelBuilder execution. The cells in the runoff from Zoun's work did not exactly overlap the cells in the precipitation grid. However, because the runoff grid used by WQModel in ModelBuilder was generated directly from the input precipitation grid, the cells overlap exactly. Because cell size for the runoff grid was so coarse, a slight shift in the cell locations resulted in enough variation in load calculations to produce different concentrations in the bay. However, even with this discrepancy, the loads matched up very well, especially given the inexact nature of nonpoint load calculations for this watershed.

	Concentration (colonies/m^3) From Zoun's Work	Concentration (colonies/m^3) From Model Builder
Upper Galveston Bay (2421)	6.27E+04	6.43E+04
Trinity Bay (2422)	4.06E+04	4.01E+04
East Bay (2423)	3.83E+04	3.82E+04
West Bay (2424)	6.05E+04	6.06E+04
Chocolate Bay (2432)	2.58E+05	2.61E+03
Lower Galveston Bay (2439)	2.39E+04	2.39E+04

Table 6.6 Bay Concentrations of Fecal Coliform

WQModel was constructed in 3 weeks, and executed in 20 minutes for Galveston Bay on an average desktop computer. The ease of execution greatly improves the efficiency of the water quality modeling algorithms developed by Zoun (2003), as those algorithms must no longer be carried out manually. In addition, WQModel may be applied to other contaminants, or other regions, simply by applying the model to different attributes on the schematic network , or a schematic network representing a different region, respectively. The modular nature of the schematic processor also permits the addition of other processing engines, to appropriately model the behavior of hydrologic features in the region of interest.

Chapter 7 Conclusions and Recommendations

At the beginning of this dissertation, three questions were posed relating to geographically integrated hydrologic modeling systems. The answers to these questions and conclusions drawn from this research are provided in this chapter, followed by recommendations for future work.

7.1 GIS ANALYSIS AND REGIONALIZATION

The first of three main questions posed in this dissertation is: *How can subregional datasets be integrated into large regional datasets?* Desktop computers do not possess the computational capabilities to calculate watershed parameters from raster datasets with a large number of cells, such as for datasets covering a large regional area. Raster analyses can still be performed for subsets of that region, but raster tools do not exist to integrate the results of those analyses for the entire region.

The Raster-Network Regionalization Technique, developed through this research, solves this problem. The Raster-Network Regionalization Technique utilizes raster-based analysis at the subregional scale and network-based attribute accumulation at the regional scale in order to process large regions in an efficient manner. First, a large region is broken up into hydrologically distinct subregions, for which the calculation of watershed parameters from raster analyses is computationally feasible. Each subregion contains a set of points, called Control Points, for which watershed parameters are desired, as well as a stream network connecting those points, and raster datasets which support watershed delineation and parameter calculation.

A unique identifier, the HydroID, is assigned to each feature in each subregion. Each HydroID is seeded with a number indicating to which subregion a given feature belongs, so that HydroIDs are unique across all subregions. For each stream feature, the HydroID of the nearest Control Point to which that stream feature drains is assigned to the JunctionID attribute of the stream feature. Watersheds are then delineated to the streams, and are automatically merged and related to their outlet Control Points through the JunctionID attribute. For each Control Point, the HydroID of the next downstream Control Point is assigned to the NextDownID attribute of the current Control Point. Thus, connectivity between watersheds and all features in the stream network is established.

The subregional datasets are then merged, and connectivity is established between subregions by updating the NextDownID attribute of the most downstream Control Point in each subregion, to point to the nearest upstream Control Point in the next downstream subregion. With the results of raster analyses stored on vector watersheds, and connectivitiy established for the vector features, the raster datasets are no longer needed. Accumulation and Consolidation tools are used to accumulate parameters from each watershed onto the stream network, and then downstream through the stream network. By calculating results for the entire region using vector data, the calculation of watershed parameters for areas covering hundreds of millions of grid cells becomes computationally feasible.

With the Raster-Network Regionalization Technique, the benefits of raster-based analysis are preserved at the subregional level, since operations such as the calculation of zonal statistics are independent of other subregions. These types of operations are useful in obtaining hydrologic information pertaining to each watershed.

Subregions are integrated into regions through the vector datasets and relationships. The stream network provides the backbone for connectivity between subregions, with watersheds being related to the network through their outlet junction on the network. The watersheds serve as **processing units** in the accumulation of parameters onto points of interest along the stream network. Thus, at the subregional scale, high-resolution raster cells are accumulated or averaged for each watershed. At the regional scale, the watersheds take the role of the grid cells, but provide a much larger "cell size" for the accumulation, which improves performance. The accuracy of the data is not compromised in this process, because the watersheds represent the maximum allowable amount of spatial averaging for each point of interest on the stream network for which watershed parameters are required.

To ensure accurate watershed delineation, this research developed the concept of delineating watersheds for source features of any geometry type. In the Raster-Network Regionalization Technique, watersheds are delineated for stream segments which drain to outlet points. Using geometries which include a larger number of equivalent raster cells provides a more effective outlet zone for capturing the flow of water as determined by the flow direction raster.

The Raster-Network Regionalization Technique has been applied to the analysis of large river basins in Texas. The technique could also be applied at a local level when high resolution data, such as LIDAR data, are available. These data are so dense that they typically preclude raster analysis over a relatively small area. Raster-Network Regionalization allows the local area to be further subdivided to accommodate raster analysis, and then integrated using relationships on vector stream network data.

In addition to the integration of subregional datasets, the network-based techniques developed for regionalization also provide useful preprocessing benefits in general. The accumulation routines can be applied at any scale. Using Arc Hydro's data structure and relationships, as well as taking advantage of vector processing techniques, provides a more integrated and efficient approach compared to raster-based preprocessing techniques used in the past.

The limits of network-based analysis were not reached in this research as far as computing time and processing power are concerned, even when analyzing river basins such as the Brazos River basin, with an east-west expanse larger than the state of Texas. Thus, with Raster-Network Regionalization and Arc Hydro, the same types of hydrologic analyses may be performed in the GIS, independent of scale.

7.2 Model Integration Through Information Exchange Points

The second of three main questions posed in this dissertation is: *How can GIS and Arc Hydro be used to integrate simulation models?* This research shows how the ModelBuilder environment in ArcGIS 9 provides the workflow sequence to process the computation of a hydrologic information system, while Arc Hydro provides the data structure through which components communicate with each other. This data structure includes *information exchange points* at which modeling computations occur, and the input and output of the models (e.g. TimeSeries, plus other attribute information). Information exchange points provide a common geospatial index through which models may communicate, using the geodatabase as the medium for data exchange. For example, a rainfall-runoff model may produce outflow hydrographs at the outlet of each watershed along the stream network, while a hydraulic model may pick up the flows from those outlet points to compute water surface elevations along the stream channel.

The integration of each model with the GIS is facilitated through the use of *interface data models*, which provide mapping between a model's data and parameters, and the corresponding GIS representation of those data and parameters. This mapping allows for efficient and clear exchange of data between the model and the GIS. An interface data is created for each simulation model to be included in the hydrologic information system. An interface data model's design is based on Arc Hydro, which provides a common data standard for water resources data. Thus, data from one simulation model may be transferred to another simulation model, by navigating from one simulation model's interface data model, through Arc Hydro, and to the other simulation model's interface data model, all within the same geodatabase. A key observation is that a simulation model no longer cares where it is getting its data from, as long as it conforms to the data structure of its interface data model.

This research developed a floodplain mapping application in the ModelBuilder environment which used the concepts of model integration through the exchange of time series data at information exchange points, and interface data models, to combine the HEC-HMS hydrologic simulation model with the HEC-RAS hydraulic simulation model, in order to convert time series of rainfall data to flood inundation polygons for Rosillo Creek in Texas. The ModelBuilder workflow sequence begins by importing NEXRAD rainfall data into the GIS, and associating that data with watersheds. These watersheds correspond to Basin objects in HMS. ModelBuilder then transfers the time series to HMS time series files, or DSS files, and calls HMS to compute outflow hydrographs for each watershed. The result are brought back into the GIS, and associated with cross section features. These cross sections in the GIS correspond to cross sections in a RAS model. ModelBuilder then transfers the time series to RAS DSS files, and calls RAS to compute water surface elevations at each cross section. These results are brought back into the GIS, and used to create a TIN of water surface elevation. The water surface is then compared to the land surface to produce flood inundation polygons.

While exchanging information through the GIS at information exchange points proved effective for the floodplain mapping application presented in this research, the methodology may not be valid for all applications. Several applications of geospatial integration with simulation models were presented in this research. An important observation about the applications is that each one supported a specific type of modeling. For example, the WRAP Hydro case study focused on water supply, while the NEXRAD to Flood Polygon application focused on floodplain mapping. Developing hydrologic information systems for a specific type of modeling frames the problem at hand and limits the scope of database and tool development to manageable levels, especially given the limited experience with ArcGIS's new ModelBuilder environment. As experience grows, and the concepts of Interface Data Modeling and hydrologic information systems mature, perhaps applications incorporating a variety of model types could be developed.

7.3 Feature-Level Behavior Implemented Using the Schematic Processor

The third of three main questions posed in this dissertation is: *How can individual features in a GIS be directly integrated with processing engines to simulate hydrologic behavior?* One limitation of ModelBuilder is that the standard ArcGIS tools are designed to operate at the dataset level, rather than the feature level. Through this research, a Schematic Network Processor was created to perform feature level operations within the ModelBuilder environment. The Schematic Processor makes use of the Arc Hydro schematic network, which represents hydrologic features in the landscape as a set of nodes, and the connectivity between those features as a set of links. The Schematic Processor associates hydrologic behavior with each schematic link or node, depending on what type of feature the link or node represents. For a given schematic feature, this behavior defines how information is received from upstream features, and what information is passed to downstream features. The Schematic Processor contains a set of default behaviors, but also allows users to implement custom behavior by creating processing engines in the form of DLLs, which the Schematic Processor calls for a given schematic feature on the fly. With hydrologic behavior defined for each feature, the Schematic Processor processes the entire schematic network, from upstream to downstream, until all features have been processed.

The modular design of the schematic processor gives the user the ability to develop processing engines geared towards solving a wide variety of water resources problems. However, the schematic processor assumes that the schematic network is dendritic, and that no backwater effects occur between schematic features. Violating these assumptions could hamper the performance of the schematic processor, or produce incorrect results.

While the schematic processor gives the user the ability to simulate hydrologic behavior at the feature level, such operations may be more inefficient than working with whole datasets. For time-invariant analysis, such as with the water quality modeling in Galveston Bay example, the computation time was very fast. But with time-varying operations, such as with the LibHydro Application, a huge cost in efficiency may be experienced, especially when compared to calling an external model directly to perform all calculations before returning control to the GIS. Therefore, the schematic processor may be a valuable tool when feature-level computations are required, although a simpler, dataset-oriented approach may be more appropriate when applicable.

7.4 RECOMMENDATIONS

Several Interface Data Models and ModelBuilder tools were developed for this research. An online catalog of Interface Data Models and ModelBuilder tools would allow users to share their work with others. Interface Data Models and ModelBuilder tools are easily documented and transported as UML diagrams and ArcGIS toolboxes, respectively. As developers created new Interface Data Models and tools, they could be added to the catalog. The catalog could function similar to how ESRI's geography network currently functions. A search for a particular simulation model would return an Interface Data Model for that model as well as tools related to that model's operation.

An important type of tool that would be very useful to others working with simulation models is the data bridge. Data bridges allow data exchange between two formats. Six bridges were developed for the NEXRAD to Flood Polygon application alone:

- ASCII NEXRAD to GIS
- GIS TimeSeries to HMS DSS
- GIS TimeSeries to RAS DSS
- DSS to GIS TimeSeries
- SDF RAS output to XML
- XML to GIS Cross Section Elevations

An online repository of Interface Data Models, bridges, and other model tools would be a significant benefit to developers of hydrologic information systems. This repository could be organized according to metadata associated with each of these resources. Each resource could then be searched just as normal datasets are searched online.

In addition to the online catalog, an improvement on this research could be made in error handling within the ModelBuilder environment. As ModelBuilder is a very new development, a robust scheme for trapping and handling errors has yet to be implemented. While some techniques and capabilities already exist which may prove useful in that arena, insufficient time was available during the course of this research to explore those avenues.

Another aspect with which this research could be improved is with scenario management. Scenario management is an important component of model simulations, and allows users to test a variety of different model configurations or scenarios to determine the best solution. Within an individual simulation model, such as HEC-HMS, utilities for scenario management are typically available. However, a scenario management utility or scheme in ModelBuilder has yet to be developed. This task may prove complicated, as a ModelBuilder model may require management for several simulation models, each with its own requirements regarding scenario management.

Carrying out these recommendations would greatly enhance the development of geographically integrated hydrologic modeling systems. The techniques and tools presented in this dissertation provide a blueprint for creating hydrologic information systems that take advantage of the latest software, computing capabilities, data availability, and programming techniques, to help model our world.

Bibliography

Abrahart, R. J., et al. "MEDRUSH - Spatial and Temporal River-basin Modelling at Scales Commensurate with Global Environmental Change." In K. Kovar and H. P. Nachtnebel (eds.), Application of Geographic Information Systems in Hydrology and Water Resources Management (pp. 47-54). Proceedings of the HydroGIS '96 Conference, Vienna, Austria, 16-19 Apr. 1996. Netherlands: Krips Repro, Meppel, 1996.

Alintex. "Alintex Script .NET Online Users Guide." 15 Oct. 2003 <http://www.alintex.com/Documentation/ascript/index.html?topicswhatisascript.htm>.

Batelaan, O., Z. Wang, and F. De Smedt. "An Adaptive GIS Toolbox for Hydrological Modelling." In K. Kovar and H. P. Nachtnebel (eds.), Application of Geographic Information Systems in Hydrology and Water Resources Management (pp. 3-9). Proceedings of the HydroGIS '96 Conference, Vienna, Austria, 16-19 Apr. 1996. Netherlands: Krips Repro, Meppel, 1996.

Bian, Ling, et al. "Integration Architecture and Internal Database for Coupling a Hydrological Model and ARC/INFO." In K. Kovar and H. P. Nachtnebel (eds.), Application of Geographic Information Systems in Hydrology and Water Resources Management (pp. 11-17). Proceedings of the HydroGIS '96 Conference, Vienna, Austria, 16-19 Apr. 1996. Netherlands: Krips Repro, Meppel, 1996.

Booch, Grady, James Rumbaugh, and Ivar Jacobson. The Unified Modeling Language User Guide. Reading: Addison-Wesley Longman, Inc., 1999.

Center for Research in Water Resources (CRWR). GIS Hydro 2003. Jul. 2003 <http://www.crwr.utexas.edu/gis/gishydro03/GISHydro2003.htm>.

Charnock, Thomas William, Peter David Hedges, and John Elgy. "Linking Multiple Process Level Models with GIS." In K. Kovar and H. P. Nachtnebel (eds.), Application of Geographic Information Systems in Hydrology and Water Resources Management (pp. 29-36). Proceedings of the HydroGIS '96 Conference, Vienna, Austria, 16-19 Apr. 1996. Netherlands: Krips Repro, Meppel, 1996.

Chow, V.T., David Maidment, and Larry Mays. Applied Hydrology. USA: McGraw-Hill, Inc., 1988.

Clark, Michael J. "Putting Water in its Place: A Perspective on GIS in Hydrology and Water Management." Hydrological Processes. 12 (1998): 821-824.

Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI). 18 Mar. 2003 <http://www.cuahsi.org/>.

Correia, Francisco Nunes, et al. "Coupling GIS with Hydrologic and Hydraulic Flood Modelling." Water Resources Management. 12 (1998): 229-249.

Correia, Francisco Nunes, et al. "Floodplain Management in Urban Developing Areas Part II. GIS-Based Flood Analysis and Urban Growth Modelling." Water Resources Management. 13 (1999): 23-37.

Crow, Susan. "Spatial Modeling Environments: Integration of GIS and Conceptual Modeling Frameworks." Proceedings of the 4th International Conference on Integrating GIS and Environmental Modeling, Banff, Canada, 2-8 Sep. 2000. <http://www.colorado.edu/Research/cires/banff/pubpapers/2/index.html>.

Davis, Kim. Object-Oriented Modeling of Rivers and Watersheds in Geographic Information Systems. Thesis. The University of Texas at Austin, 1999. CRWR Online Report 00-7. <http://www.crwr.utexas.edu/reports/2000/rpt00-7.shtml>.

de Roo, A. P. J. "Modelling Runoff and Sediment Transport in Catchments Using GIS." Hydrological Processes. 12 (1998): 905-922.

Environmental Protection Agency. "BASINS." 4 Nov. 2002 <http://www.epa.gov/OST/BASINS/>.

Environmental Protection Agency. "BASINS 3.0 User's Manual." EPA Office of Water, Jun. 2001.

Feng, Chen-Chieh. "Open Hydrologic Model for Facilitating GIS and Hydrologic Model Interoperability." Proceedings of the 4th International Conference on Integrating GIS and Environmental Modeling, Banff, Canada, 2-8 Sep.

2000.
<http://www.colorado.edu/Research/cires/banff/pubpapers/6/index.html>.

Figurski, Melissa J. GIS Algorithms for Large Watersheds with Non-contributing Areas. Thesis. The University of Texas at Austin, 2001. CRWR Online Report 01-7. <http://www.crwr.utexas.edu/reports/2001/rpt01-7.shtml>.

Gao, Xiaogang, Soroosh Sorooshian, and David C. Goodrich. "Linkage of a GIS to a Distributed Rainfall-Runoff Model." In M. F. Goodchild, B. O. Parks, and Louis T. Steyaert (eds.), Environmental Modeling with GIS (pp. 182-187). New York: Oxford University Press, Inc., 1993.

Goodchild, Michael F. "The State of GIS for Environmental Problem-Solving." In M. F. Goodchild, B. O. Parks, and Louis T. Steyaert (eds.), Environmental Modeling with GIS (pp. 8-15). New York: Oxford University Press, Inc., 1993.

Goodchild, M.F., and K.K. Kemp, eds. NCGIA Core Curriculum in GIS. National Center for Geographic Information and Analysis, University of California, Santa Barbara, CA. 1990 <http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u23.html#UNIT23>.

Gopalan, Hema. WRAPHydro Data Model: Finding Input Parameters for the Water Rights Analysis Package. Thesis. The University of Texas at Austin, 2003. CRWR Online Report 03-3 . <http://www.crwr.utexas.edu/reports/2003/rpt03-3.shtml>.

Graham, Wendy, et al. "Hydrologic Information Systems." CUAHSI White Paper. 7 Oct. 2002 <http://www.iihr.uiowa.edu/~cuahsi/his/documents/index.html>.

HarmonIT. "IT Frameworks – HarmonIT." 29 Jan. 2003 <http://www.harmonit.org/>.

HEC. LibHydro Users Manual. Davis: Hydraulic Engineering Center, 1995.

HEC. HEC-HMS. Hydrologic Modeling System, Version 2.1. 17 Apr. 2001 <http://www.hec.usace.army.mil/software/software_distrib/hec-hms/hechmsprogram.html>.

Hartley, Tim. GUI Design Fundamentals Learning Guide. Phoenix: ComputerPREP, 1998.

Hickman, Charley. "National Hydrography Dataset." 7 Nov. 2002 <http://www.state.mi.us/dmb/mic/gis/nhd.htm>.

Hudgens, Bradley T., and David R. Maidment. Geospatial Data in Water Availability Modeling. Thesis. The University of Texas at Austin, 1999. CRWR Online Report 99-4 . <http://www.crwr.utexas.edu/reports/1999/rpt99-4.shtml>.

Hutchings, Chris, et al. "State of the Art Review: Work Package 1." IT Frameworks (HarmonIT), Sep. 2002.

Kidner, David B., and Derek H. Smith. "Advances in the Data Compression of Digital Elevation Models." Computers and Geosciences. 29 (2003): 985-1002.

Kopp, Stephen M. "Linking GIS and Hydrological Models: Where We Have Been, Where We Are Going." In K. Kovar and H. P. Nachtnebel (eds.), Application of Geographic Information Systems in Hydrology and Water Resources Management (pp. 133-140). Proceedings of the HydroGIS '96 Conference, Vienna, Austria, 16-19 Apr. 1996. Netherlands: Krips Repro, Meppel, 1996.

Koussis, Antonis D., et al. "Flood Forecasts for Urban Basin with Integrated Hydro-Meteorological Model." Journal of Hydrologic Engineering. 8 (2003) 1-11.

Loague, K., and D. L. Corwin. "Regional-scale Assessment of Non-point Source Groundwater Contamination." Hydrological Processes. 12 (1998): 957-966.

Maidment, David R., ed. Arc Hydro: GIS for Water Resources. Redlands: ESRI Press, 2002.

Mason, David. An Analysis of a Methodology for Generating Watershed Parameters using GIS. Thesis. The University of Texas at Austin, 2000. CRWR Online Report 00-3. <http://www.crwr.utexas.edu/reports/2000/rpt00-3.shtml>.

Meyer, Bertrand. Object-Oriented Software Construction. Hertfordshire: Prentice Hall, 1997.

Ogden, Fred L., et al. "GIS and Distributed Watershed Models. II: Modules, Interfaces, and Models." Journal of Hydrologic Engineering. 6 (2001): 515-523.

Olivera, Francisco. "BASINS Training Course." CD-ROM, Mar. 2002.

Osborne, Katherine G. A Water Quality GIS Tool for the City of Austin Incorporating Nonpoint Sources and Best Management Practices. Thesis. The University of Texas at Austin, 2000. CRWR Online Report 00-10. <http://www.crwr.utexas.edu/reports/2000/rpt00-10.shtml>.

Prisloe Jr., Michael P., et al. "A Simple GIS-based Model to Characterize Water Quality in Connecticut Watersheds." Proceedings of the 4th International Conference on Integrating GIS and Environmental Modeling, Banff, Canada, 2-8 Sep. 2000. <http://www.colorado.edu/Research/cires/banff/pubpapers/11/index.html>.

Pullar, David, and Darren Springer. "Towards Integrating GIS and Catchment Models." Environmental Modelling & Software. 15 (2000): 451-459.

Robbins, Clarence, and Stephen P. Phipps. " GIS/Water Resources Tools for Performing Floodplain Management Modeling Analysis." Proceedings of the AWRA Symposium on GIS and Water Resources, Ft. Lauderdale, FL, 22-26 Sep. 1996. <http://www.awra.org/proceedings/gis32/woolprt3/index.html>.

Roberts, A. M., and R. V. Moore. "Data and Databases for Decision Support." Hydrological Processes. 12 (1998): 823-834.

Rumbaugh, James, et al. Object-Oriented Modeling and Design. Englewood Cliffs: Prentice Hall, 1991.

Schumann, A. H., and R. Funke. "GIS-based Components for Rainfall-runoff Models." In K. Kovar and H. P. Nachtnebel (eds.), Application of Geographic Information Systems in Hydrology and Water Resources Management (pp. 477-484). Proceedings of the HydroGIS '96 Conference, Vienna, Austria, 16-19 Apr. 1996. Netherlands: Krips Repro, Meppel, 1996.

Stone, Sarah A. Geospatial Database and Preliminary Flood Hydrology Model for the Lower Colorado Basin. Thesis. The University of Texas at Austin,

2001. CRWR Online Report 01-4. <http://www.crwr.utexas.edu/reports/2001/rpt01-4.shtml>.

Storck, P., et al. "Application of a GIS-based Distributed Hydrology Model for Prediction of Forest Harvest Effects on Peak Streamflow in the Pacific Northwest." Hydrological Processes. 12 (1998): 889-904.

Streit, Ulrich, and Hans Kleeberg. "GIS-based Regionalization in Hydrology: German Priority Programme on Spatial Transfer of Hydrological Information." In K. Kovar and H. P. Nachtnebel (eds.), Application of Geographic Information Systems in Hydrology and Water Resources Management (pp. 485-491). Proceedings of the HydroGIS '96 Conference, Vienna, Austria, 16-19 Apr. 1996. Netherlands: Krips Repro, Meppel, 1996.

Takamatsu, Masatsugu. "LibHydro." 17 Oct. 2003. <http://civilu.ce.utexas.edu/stu/takamam/lh_home.htm>.

United States Geological Survey. "MMS Welcome Page." 5 Nov. 2002a. <http://wwwbrr.cr.usgs.gov/mms/>.

United States Geological Survey. "NWISWeb Data for the Nation." 31 Oct. 2002b <http://waterdata.usgs.gov/nwis>.

Vieux, Baxter E. Distributed Hydrologic Modeling Using GIS. Dordrecht: Kluwer Academic Publishers, 2001.

Whiteaker, Tim. A Prototype Toolset for the ArcGIS Hydro Data Model. Thesis. The University of Texas at Austin, 2001. CRWR Online Report 01-6. <http://www.crwr.utexas.edu/reports/2001/rpt01-6.shtml>.

Whiteaker, Tim, and Jon Goodall. "Water Quality Modeling in Galveston Bay with Model Builder." Proceedings of ESRI International User Conference. San Diego, CA. 2003.

Wurbs, R. A. "Reference and Users Manual for the Water Rights Analysis Package (WRAP)." Texas Water Resources Institute, Jul. 2001.

Yoon, Jaewon. "Watershed-Scale Nonpoint Source Pollution Modeling and Decision Support System Based on a Model-GIS-RDBMS Linkage." Proceedings of the AWRA Symposium on GIS and Water Resources, Ft. Lauderdale, FL, 22-26 Sep. 1996. <http://www.awra.org/proceedings/gis32/jyoon/index.html>.

Zoun, Reem Jihan. Estimation of Fecal Coliform Loadings to Galveston Bay. Thesis. The University of Texas at Austin, 2003. CRWR Online Report 03-5. <http://www.crwr.utexas.edu/reports/2003/rpt03-5.shtml>.

Vita

Timothy Lee Whiteaker was born in Cookeville, Tennessee on April 3, 1976, the son of Leigh Whiteaker and Charles Ed Whiteaker. After completing his work at Cumberland County High School, Crossville, Tennessee, in 1994, he entered Tennessee Technological University in Cookeville, Tennessee. He received the degree of Bachelor of Science in Civil and Environmental Engineering from Tennessee Technological University in May, 1999. In September, 1999, he entered The Graduate School at The University of Texas at Austin. He received the degree of Master of Science in Engineering from The University of Texas at Austin in August, 2001.

Permanent address: 523 Madison Ave.
Sparta, TN 38583

This dissertation was typed by the author.